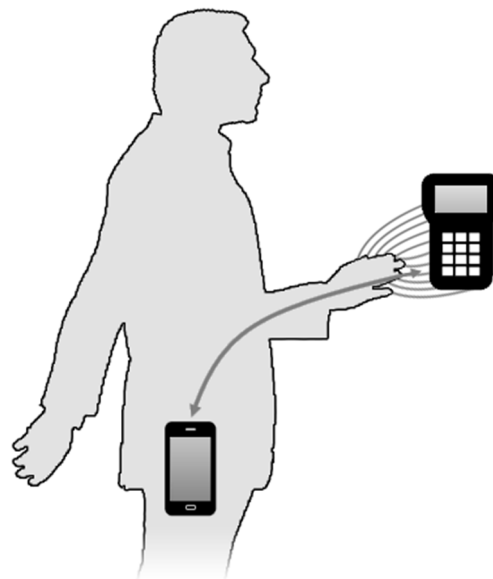




ANDRÉ FILIPE  
MARANTE FONSECA

## SISTEMA DE PAGAMENTO SEGURO USANDO COMUNICAÇÃO ATRAVÉS DO CORPO HUMANO







**ANDRÉ FILIPE  
MARANTE FONSECA**

## **SISTEMA DE PAGAMENTO SEGURO USANDO COMUNICAÇÃO ATRAVÉS DO CORPO HUMANO**

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia Electrónica e Telecomunicações, realizada sob a orientação científica do Professor Doutor Rui Manuel Escadas Ramos Martins, Professor Auxiliar do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro, e a co-orientação do Professor Doutor André Ventura da Cruz Marnoto Zúquete, Professor Auxiliar do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro.





## o júri

presidente

**Professor Doutor Armando Carlos Domingues da Rocha**

Professor auxiliar do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro

vogais

**Professor Doutor Valter Filipe Miranda Castelão da Silva**

Professor adjunto da Universidade de Aveiro (arguente principal)

**Professor Doutor Rui Manuel Escadas Ramos Martins**

Professor auxiliar do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro (orientador)



## agradecimentos

Dedico esta dissertação de mestrado à minha família, particularmente aos meus pais, ao meu irmão, aos meus avós e à minha namorada.

Extendo ainda esta dedicatória a todas as restantes pessoas que mais estimo, incluindo todos os meus amigos.

Expresso o meu profundo agradecimento ao meu orientador Rui Escadas por todo o apoio prestado, pelas dicas e contribuições cruciais, pelo conhecimento transmitido, pela confiança na minha pessoa, pela simpatia, pela boa-disposição contagiante e pela amizade.

Deixo também o meu especial agradecimento ao colega Bruno Pereira por todo o trabalho conjunto realizado no âmbito do desenvolvimento do sistema de comunicação, pela entajuda, pela cumplicidade, pela motivação transmitida e pela amizade.

Por fim, agradeço ao meu co-orientador André Zúquete pelas dicas fornecidas durante a fase inicial deste projecto e aos colegas Tiago Gonçalves e Rómulo Antão por toda a ajuda proporcionada durante o desenvolvimento e assemblagem das placas de circuito impresso.



## palavras-chave

sistema de pagamento via BCC, comunicação acoplada ao corpo humano, acoplamento capacitivo, corpo humano, computação móvel, pagamento móvel, pagamento electrónico, contactless BCC, transceiver, telemóvel, smartphone, Android, BAN, body area network, HAN, human area network, BCC, body coupled communication, HBC, human body communication, NFC, near field communication

## resumo

Actualmente existem diversos meios de pagamentos electrónicos, os quais têm vindo a ser cada vez mais utilizados a nível mundial. Com a evolução da tecnologia e dos dispositivos móveis no decorrer da última década, surgiram novas tendências para os sistemas de pagamento, nomeadamente a realização de pagamentos móveis através de telemóveis ou smartphones. O sistema de pagamento via NFC, o qual tem vindo a afirmar-se cada vez mais neste paradigma, é um sistema que permite realizar pagamentos móveis por proximidade e de forma segura utilizando um smartphone equipado com a devida tecnologia. Contudo, é necessário que o utilizador aproxime o seu dispositivo do terminal de pagamentos durante todo o processo de pagamento, algo que poderá ser incómodo ou mesmo inconveniente em determinadas situações.

O sistema de pagamento via BCC, desenvolvido neste projecto de dissertação de mestrado, procura oferecer uma nova solução no âmbito dos sistemas de pagamento móveis. O mesmo utiliza o corpo humano como meio de comunicação, fazendo uso de tecnologia que permite a transmissão e a recepção da informação por acoplamento capacitivo ao corpo humano. Devido ao confinamento da comunicação no corpo humano, com este sistema aumenta-se a segurança de um pagamento e eliminam-se os incómodos e inconveniências do sistema de pagamento via NFC.



**keywords**

BCC payment system, capacitive, mobile phone, mobile computing, mobile payment, electronic payment, contactless BCC, transceiver, smartphone, Android, BAN, body area network, HAN, human area network, BCC, body coupled communication, HBC, human body communication, NFC, near field communication

**abstract**

There are currently various means of electronic payments, which have been increasingly used worldwide. With the evolution of technology and mobile devices over the last decade, new trends in payment systems have emerged, namely the realization of mobile payments using mobile phones or smartphones. The NFC payment system, which has been increasingly asserting itself in this paradigm, is a system that allows performing mobile payments by proximity and securely using an equipped smartphone with the proper technology. However, it is necessary that the user approaches the mobile device to the POS terminal throughout the whole payment process, something that can be uncomfortable or inconvenient in certain situations. The BCC payment system, developed in this Master's dissertation project, seeks to offer a new solution in the framework of mobile payment systems. It uses the human body as a communication medium, making use of technology that allows the transmission and reception of information through capacitive coupling to the human body. Due to the communication confinement in the human body, this system increases the security of a payment and eliminates the hassle and inconveniences of the NFC payment system.





# Índice

<b>1</b>	<b>Introdução .....</b>	<b>1</b>
1.1	Motivação .....	1
1.2	Objectivos .....	1
1.3	Metodologia .....	2
1.4	Estrutura da dissertação .....	2
<b>2</b>	<b>Estado da arte.....</b>	<b>5</b>
2.1	Sumário.....	5
2.2	Introdução aos conceitos de PAN, BAN e BCC .....	5
2.2.1	Personal Area Networks (PAN) e Body Area Networks (BAN) .....	5
2.2.2	Body Coupled Communication (BCC).....	10
2.3	Body Coupled Communication (BCC) .....	14
2.3.1	Introdução.....	14
2.3.2	Abordagens em BCC.....	14
2.3.3	Alcance e distribuição do campo eléctrico.....	17
2.3.4	Frequência portadora ideal .....	20
2.3.5	Eficiência da transmissão .....	22
2.3.6	Consumo de energia .....	23
2.3.7	Modelo simplificado para BCC .....	25
2.4	Sistema BCC da Universidade de Aveiro.....	27
2.4.1	Hugo Barreira, Universidade de Aveiro, 2009 [33] .....	27
2.4.2	José Reis, Universidade de Aveiro, 2010 [34] .....	28
2.4.3	Humberto Castanheira, Universidade de Aveiro, 2012 [35].....	29
2.4.4	José Lima, Universidade de Aveiro, 2013 [36].....	29
2.5	Sistemas de pagamento.....	30
2.5.1	Introdução.....	30
2.5.2	Sistemas de pagamento móveis .....	32
2.5.3	Carteira digital.....	33
2.5.4	Premium SMS.....	33
2.5.5	Direct Mobile Billing.....	34
2.5.6	Mobile Web.....	34
2.5.7	Contactless NFC .....	35
<b>3</b>	<b>Sistema de comunicação.....</b>	<b>41</b>
3.1	Sumário.....	41
3.2	Arquitectura do sistema de comunicação.....	41
3.3	Microcontrolador.....	45

3.4	Sincronismo UART .....	45
3.5	Emissor .....	46
3.5.1	Modulador FSK .....	46
3.5.2	IF up-conversion mixer .....	48
3.5.3	RF up-conversion mixer .....	52
3.5.4	Amplificador sintonizado .....	53
3.6	Receptor .....	56
3.6.1	Amplificador sintonizado .....	56
3.6.2	RF down-conversion mixer .....	59
3.6.3	Amplificador FI .....	62
3.6.4	Limitador .....	62
3.6.5	Desmodulador FSK .....	62
3.6.6	Amplificador de banda-base .....	66
3.6.7	Comparador .....	67
3.7	Alimentação .....	69
3.7.1	Regulador de tensão .....	69
3.7.2	Controlador de gestão de carga .....	69
3.7.3	Multiplicador de tensão .....	70
3.8	Placa de circuito impresso .....	70
3.9	Considerações finais .....	72
<b>4</b>	<b>Sistema de pagamento .....</b>	<b>73</b>
4.1	Sumário .....	73
4.2	Conceito e arquitectura do sistema de pagamento .....	73
4.3	Interface Android .....	80
4.3.1	Interface de interligação .....	80
4.3.2	Terminal de pagamentos .....	83
4.3.3	Carteira digital .....	87
4.4	Software e estruturação dos dados .....	90
4.4.1	Entidade vendedora .....	90
4.4.2	Utilizador (cliente) .....	91
4.4.3	Pagamento .....	91
4.4.4	IOIO-OTG .....	92
4.4.5	BCC Terminal .....	93
4.4.6	BCC Wallet .....	104
4.5	Considerações finais .....	117
<b>5</b>	<b>Resultados .....</b>	<b>119</b>
5.1	Sumário .....	119
5.2	Sistema de comunicação .....	119
5.2.1	Transceiver implementado .....	119
5.2.2	Bloco emissor do transceiver .....	120
5.2.3	Bloco receptor do transceiver .....	124
5.2.4	Alcance do transceiver .....	129
5.2.5	Consumo do transceiver .....	131
5.3	Sistema de pagamento .....	131
5.3.1	Modelação inicial (sem transceivers) .....	131

5.3.2	Sistema final (com transceivers) .....	137
5.4	Considerações finais .....	138
<b>6</b>	<b>Conclusão e trabalho futuro .....</b>	<b>139</b>
6.1	Sumário.....	139
6.2	Conclusão.....	139
6.3	Trabalho futuro .....	140
6.3.1	Transceiver low-power .....	140
6.3.2	Sistema de pagamento .....	140



# Índice de figuras

Figura 2.1. Thomas Zimmerman [2].	5
Figura 2.2. Protótipo desenvolvido por Thomas Zimmerman [3].	6
Figura 2.3. Experiência realizada por Thomas Zimmerman [4].	7
Figura 2.4. Exemplo de uma PAN.	7
Figura 2.5. Exemplo de uma BAN [6].	8
Figura 2.6. Exemplos práticos de interação entre PAN e BAN [7] e [8].	8
Figura 2.7. Posicionamento de uma PAN e de uma BAN no paradigma das redes sem fios ( <i>wireless</i> ) [9].	9
Figura 2.8. Algumas limitações das tecnologias utilizadas nas PAN e BAN actuais comparativamente às características de uma BAN ideal [10].	9
Figura 2.9. Gráfico comparativo das várias tecnologias sem fios ( <i>wireless</i> ) utilizadas nas PAN e BAN actuais (taxa de transmissão vs potência consumida) [13].	12
Figura 2.10. Gráfico comparativo das várias tecnologias sem fios ( <i>wireless</i> ) utilizadas nas PAN e BAN actuais (distância de propagação no ar vs taxa de transmissão) [14].	12
Figura 2.11. Exemplos de aplicações que podem utilizar BCC [6].	13
Figura 2.12. Novas capacidades das BAN com a introdução de tecnologias BCC [15].	13
Figura 2.13. Acoplamento galvânico. Baseado em [19] e [21].	15
Figura 2.14. Acoplamento capacitivo [14].	16
Figura 2.15. Estruturas horizontal e vertical dos eléctrodos [16].	16
Figura 2.16. Orientações para a estrutura horizontal dos eléctrodos [16].	16
Figura 2.17. Regiões <i>near-field</i> e <i>far-field</i> [23].	17
Figura 2.18. Modelo de três regiões <i>near-field</i> e <i>far-field</i> [24].	17
Figura 2.19. Distribuições do campo eléctrico para várias frequências. (a) 3 MHz. (b) 30 MHz. (c) 300 MHz. (d) 3 GHz. [36].	18
Figura 2.20. Distribuição do campo eléctrico para diferentes frequências [26].	19
Figura 2.21. Ganho do sinal recebido para os dois casos [27].	20
Figura 2.22. Configuração utilizada nas experiências realizadas em [28].	21
Figura 2.23. Comparação entre as impedâncias obtidas no contacto do corpo humano com eléctrodos de diferentes metais [28].	21
Figura 2.24. Comparação entre os ganhos obtidos para diferentes frequências portadoras utilizando o corpo humano e o ar como canais de transmissão [28].	22
Figura 2.25. Constelações obtidas em [29].	23
Figura 2.26. Algumas soluções BCC apresentadas nos últimos anos [31].	24
Figura 2.27. Gráfico comparativo do consumo de energia de microprocessadores utilizados em BCC e <i>transceivers</i> utilizados em tecnologias <i>wireless</i> comuns [7].	25
Figura 2.28. Abordagem considerada em [32].	25

Figura 2.29. Modelo simplificado [32]. (a) Sistema de três condutores. (b) Equivalente eléctrico do sistema.....	26
Figura 2.30. Sistema construído e sistema simulado em [32].....	26
Figura 2.31. Resultados obtidos em [32].....	27
Figura 2.32. Diagrama de blocos do sistema implementado por Hugo Barreira [33]. ....	28
Figura 2.33. Protótipo construído por Hugo Barreira [33].....	28
Figura 2.34. Protótipo construído por Humberto Castanheira [35]. ....	29
Figura 2.35. <i>Transceiver</i> construído por José Lima e respectivas PCBs. À esquerda: <i>top layer</i> . À direita: <i>bottom layer</i> [36].....	30
Figura 2.36. Vários formatos de dinheiro existentes no mundo actual. ....	31
Figura 2.37. Bitcoin.....	31
Figura 2.38. Meios/dispositivos de pagamento mais utilizados a nível mundial.....	32
Figura 2.39. NFC. ....	35
Figura 2.40. Formas de pagamento via <i>contactless</i> NFC. ....	36
Figura 2.41. Modo de operação activo em NFC. Baseado em [46]. ....	38
Figura 2.42. Modo de operação passivo em NFC. Baseado em [46].....	38
Figura 2.43. Modos de comunicação em NFC [47]. ....	39
Figura 2.44. Funcionamento do modo Card Emulation. Baseado em [48].....	40
Figura 2.45. Exemplo genérico de uma comunicação entre uma aplicação (MIDlet) e um <i>smart-card</i> (Security Element) [48]. ....	40
Figura 3.1. Arquitectura básica de um sistema de comunicação. ....	42
Figura 3.2. Arquitectura do sistema de comunicação implementado.....	43
Figura 3.3. Trama UART configurada. ....	46
Figura 3.4. Diagrama de blocos do emissor. ....	46
Figura 3.5. Modulador FSK tradicional. Baseado em [35].....	47
Figura 3.6. Diagrama de blocos do modulador FSK implementado. ....	47
Figura 3.7. Exemplo da modulação FSK implementada.....	48
Figura 3.8. Diagrama de blocos do <i>mixer</i> NXP SA602A [49]. ....	49
Figura 3.9. Atenuador e filtro passa-banda à entrada do <i>mixer</i> NXP SA602A.....	49
Figura 3.10. Circuito que realiza a conversão do sinal BFSK para o sinal IF BFSK e a posterior filtragem na banda de interesse. ....	50
Figura 3.11. Processo de IF <i>up-conversion</i> efectuado pelo <i>mixer</i> NXP SA602A. ....	50
Figura 3.12. Resposta em frequência do filtro passa-banda Murata SFEC10M7 [50].....	51
Figura 3.13. Sinal IF BFSK filtrado.....	52
Figura 3.14. Circuito que realiza a conversão do sinal IF BFSK para o sinal RF BFSK.....	53
Figura 3.15. Amplificador sintonizado do andar de saída do emissor.....	54
Figura 3.16. Ilustração da resposta em frequência do amplificador sintonizado do emissor consoante o factor de qualidade (Q) da malha LC. ....	54
Figura 3.17. Resposta em frequência do amplificador sintonizado do emissor [35].....	55
Figura 3.18. Resposta em frequência do amplificador sintonizado do emissor [36].....	55
Figura 3.19. Diagrama de blocos do receptor. ....	56
Figura 3.20. Amplificador sintonizado do andar de entrada do receptor.....	57
Figura 3.21. Resposta em frequência do amplificador sintonizado do receptor [35].....	58
Figura 3.22. Resposta em frequência do amplificador sintonizado do receptor [36].....	58
Figura 3.23. Diagrama de blocos do NXP SA636 [51]. ....	59
Figura 3.24. Processo de RF <i>down-conversion</i> (caso <i>low-side</i> ). ....	60
Figura 3.25. Circuito do andar de frequência intermédia (FI) do receptor.....	61

Figura 3.26. Aspecto básico de um circuito detector de quadratura. Baseado em [35].	63
Figura 3.27. Detector de quadratura utilizado. Baseado em [51].	63
Figura 3.28. Geração da tensão de referência do <i>varicap</i> (sinal VTUNNING).	64
Figura 3.29. Ilustração da variação da tensão de referência do <i>varicap</i> consoante a variação do <i>duty-cycle</i> do sinal PWM gerado pelo microcontrolador.	64
Figura 3.30. Simulação da resposta em frequência do detector de quadratura utilizado [35].	65
Figura 3.31. Circuito do amplificador de banda-base.	66
Figura 3.32. Diagrama de blocos do circuito de decisão utilizado para a conversão do sinal desmodulado num sinal digital binário.	67
Figura 3.33. Diagrama de blocos do módulo de triplo comparador do microcontrolador [52].	68
Figura 3.34. Diagrama de blocos do gerador de tensões de referência ( <i>resistor ladder</i> ) do microcontrolador [52].	68
Figura 3.35. Configuração utilizada para o Texas Instruments TLV70030. Baseado em [53].	69
Figura 3.36. Configuração utilizada para o Microchip MCP73831. Baseado em [54].	69
Figura 3.37. Configuração utilizada para o Texas Instruments REG710 [55].	70
Figura 3.38. PCB desenvolvida. (a) Camada superior ( <i>top layer</i> ). (b) Camada inferior ( <i>bottom layer</i> ). (c) Camada superior intermédia. (d) Camada inferior intermédia.	71
Figura 4.1. Ilustração do sistema de pagamento concebido.	74
Figura 4.2. Número de <i>smartphones</i> vendidos em todo o mundo nos últimos anos para diversos sistemas operativos móveis [56].	74
Figura 4.3. Placa de desenvolvimento IOIO-OTG.	75
Figura 4.4. Arquitectura simplificada do sistema de pagamento implementado.	76
Figura 4.5. Arquitectura detalhada do sistema de pagamento implementado.	76
Figura 4.6. Arquitectura ideal para o sistema de pagamento concebido.	77
Figura 4.7. Comunicação <i>smartphone</i> - terminal.	78
Figura 4.8. Comunicação terminal - <i>smartphone</i> .	78
Figura 4.9. Diagrama de blocos da interface de interligação implementada.	80
Figura 4.10. Terminações USB utilizadas para a interligação entre os <i>smartphones</i> e as placas IOIO-OTG.	81
Figura 4.11. Esquemático eléctrico parcial das placas IOIO-OTG utilizadas.	82
Figura 4.12. Fluxograma do funcionamento da aplicação BCC Terminal.	84
Figura 4.13. Diagrama com as janelas da aplicação BCC Terminal.	86
Figura 4.14. Fluxograma do funcionamento da aplicação BCC Wallet.	87
Figura 4.15. Diagrama com as janelas da aplicação BCC Wallet.	89
Figura 4.16. Diagrama ilustrativo de como é processado um pagamento pela aplicação BCC Wallet (ponto de vista do utilizador).	90
Figura 4.17. Entidade vendedora.	90
Figura 4.18. Utilizador (cliente).	91
Figura 4.19. Pagamento.	91
Figura 4.20. Trama enviada pelo <i>smartphone</i> do utilizador (identificação).	96
Figura 4.21. Trama enviada pelo terminal de pagamentos (identificação).	99
Figura 4.22. Trama enviada pelo <i>smartphone</i> do utilizador (confirmação).	99
Figura 4.23. Trama enviada pelo terminal de pagamentos (conclusão).	102

Figura 5.1. (a) Camada superior ( <i>top layer</i> ) do transceiver. (b) Camada inferior ( <i>bottom layer</i> ) do transceiver. (c) Eléctrodo do transceiver. ....	120
Figura 5.2. (a) Sinais à entrada e saída do modulador FSK. (b) Sinais à entrada e saída do modulador FSK (sinais ampliados). (c) Espectro do sinal modulado. ....	121
Figura 5.3. (a) Sinal à entrada do IF <i>up-conversion mixer</i> . (b) Espectro do sinal à saída do IF <i>up-conversion mixer</i> . ....	122
Figura 5.4. Espectro do sinal à saída do filtro passa-banda do emissor. ....	122
Figura 5.5. Espectro do sinal à saída do RF <i>up-conversion mixer</i> . ....	123
Figura 5.6. (a) Sinal à saída do amplificador sintonizado do emissor. (b) Espectro do sinal à saída do amplificador sintonizado do emissor. ....	123
Figura 5.7. (a) Sinal à entrada do amplificador sintonizado do receptor. (b) Espectro do sinal à entrada do amplificador sintonizado do receptor. (c) Sinal à saída do amplificador sintonizado do receptor. (d) Espectro do sinal à saída do amplificador sintonizado do receptor. ....	125
Figura 5.8. (a) Sinal à saída do RF <i>down-conversion mixer</i> . (b) Espectro do sinal à saída do RF <i>down-conversion mixer</i> . ....	125
Figura 5.9. (a) Sinal à entrada do amplificador FI. (b) Espectro do sinal à entrada do amplificador FI. (c) Sinal à saída do amplificador FI. (d) Espectro do sinal à saída do amplificador FI. ....	126
Figura 5.10. (a) Sinal à entrada do limitador. (b) Espectro do sinal à entrada do limitador. (c) Sinal à saída do limitador. ....	127
Figura 5.11. (a) Sinal desmodulado à saída do desmodulador FSK. (b) Sinal desmodulado amplificado à saída do amplificador de banda-base do receptor. ....	128
Figura 5.12. Sinal digital binário à saída do comparador. ....	128
Figura 5.13. Testes efectuados. (a) Teste 1. (b) Teste 2. (c) Teste 3. (d) Teste 4. (e) Teste 5. (f) Teste 6. ....	129
Figura 5.14. Nível da RSSI para diferentes distâncias entre o emissor e o receptor para o caso em que o canal de transmissão é uma superfície não condutora (Teste 1). ....	130
Figura 5.15. Configuração utilizada para o teste em <i>loopback</i> . ....	132
Figura 5.16. Trama enviada (teste em <i>loopback</i> ). ....	133
Figura 5.17. Trama recebida (teste em <i>loopback</i> ). ....	133
Figura 5.18. Configuração utilizada para o teste com ligação <i>shunt</i> . ....	134
Figura 5.19. Configuração final do sistema de pagamento implementado. ....	138
Figura A.1. Microcontrolador. ....	148
Figura A.2. Controlador de carga, regulador de tensão e multiplicador de tensão. ....	149
Figura A.3. IF <i>up-conversion mixer</i> e filtro passa-banda. ....	149
Figura A.4. RF <i>up-conversion mixer</i> . ....	149
Figura A.5. Amplificador sintonizado do emissor. ....	150
Figura A.6. Amplificador sintonizado do receptor. ....	150
Figura A.7. Amplificador de banda-base do receptor. ....	150
Figura A.8. Andar de FI do receptor. ....	151
Figura B.1. Esquemático eléctrico da placa IOIO-OTG. ....	154



# Índice de tabelas

Tabela 2.1. Implementações BCC realizadas nos últimos anos por vários investigadores de todo o mundo [12].....	11
Tabela 2.2. Comparação qualitativa da tecnologia BCC relativamente a outros tipos de comunicação. Baseado em [18].....	14
Tabela 2.3. Eficiência de radiação [36]. .....	18
Tabela 2.4. Atenuação do sinal recebido para as diferentes distâncias.....	20
Tabela 2.5. Principais diferenças entre o NFC e o HF RFID.....	37
Tabela 2.6. Características da tecnologia NFC. ....	37
Tabela 2.7. Codificações e modulações utilizadas em NFC. ....	37
Tabela 4.1. Criação de uma subclasse da IOIOActivity.....	92
Tabela 4.2. Criação de uma subclasse da BaseIOIOLooper.....	93
Tabela 4.3. Criação da thread para o processamento relativo à placa IOIO-OTG.....	93
Tabela 4.4. Temporizador para cancelamento do processo de pagamento.....	95
Tabela 4.5. Configuração da trama de dados UART para a placa IOIO-OTG.....	95
Tabela 4.6. Detecção do utilizador (ConfirmActivity). ....	96
Tabela 4.7. Identificação do utilizador (ConfirmActivity).....	97
Tabela 4.8. Detecção do utilizador (PayActivity). ....	99
Tabela 4.9. Identificação do utilizador (PayActivity). ....	100
Tabela 4.10. Conclusão do processo de pagamento (PayActivity). ....	101
Tabela 4.11. Apresentação da factura no ecrã do terminal (PayActivity). ....	103
Tabela 4.12. AndroidManifest da aplicação BCC Terminal. ....	104
Tabela 4.13. Activação/desactivação do AppService (MainActivity).....	106
Tabela 4.14. Lista dos pagamentos efectuados (MainActivity). ....	107
Tabela 4.15. Esvaziamento da lista dos pagamentos efectuados (MainActivity).....	108
Tabela 4.16. Apresentação da factura de um pagamento (PayActivity). ....	109
Tabela 4.17. Inicialização do processo de pagamento (AppService). ....	111
Tabela 4.18. Conclusão do processo de pagamento (AppService).....	113
Tabela 4.19. Classe Payment. ....	114
Tabela 4.20. Classe DBHelper. ....	116
Tabela 4.21. AndroidManifest da aplicação BCC Wallet. ....	117
Tabela 5.1. Nível da RSSI para os casos em que o canal de transmissão é o corpo humano (Testes 2, 3, 4, 5 e 6).....	130
Tabela 5.2. Consumo dos modos emissor e receptor do <i>transceiver</i> . ....	131
Tabela 5.3. Resultados obtidos para o teste com ligação <i>shunt</i> . ....	137
Tabela 5.4. Comparação entre os resultados obtidos para os dois testes efectuados. ....	137



# Lista de acrónimos

<b>ADC</b>	Analog-to-Digital Converter
<b>ADT</b>	Android Development Tools
<b>AM</b>	Amplitude Modulation
<b>APDU</b>	Application Protocol Data Unit
<b>ASCII</b>	American Standard Code for Information Interchange
<b>API</b>	Application Programming Interface
<b>ASK</b>	Amplitude-Shift Keying
<b>ATM</b>	Automatic Teller Machine
<b>BAN</b>	Body Area Network
<b>BFSK</b>	Binary Frequency-Shift Keying
<b>BJT</b>	Bipolar Junction Transistor
<b>BPSK</b>	Binary Phase-Shift Keying
<b>BCC</b>	Body Coupled Communication
<b>DC</b>	Direct Current
<b>DSSS</b>	Direct Sequence Spread Spectrum
<b>ECG</b>	Eletrocardiograma
<b>EUA</b>	Estados Unidos da América
<b>FM</b>	Frequency Modulation
<b>FI</b>	Frequência Intermédia
<b>FSK</b>	Frequency-Shift Keying
<b>GPIO</b>	General Purpose Input/Output
<b>HAN</b>	Human Area Networks
<b>HBC</b>	Human Body Communication
<b>HF</b>	High Frequency
<b>HTML</b>	HyperText Markup Language
<b>I2C</b>	Inter-Integrated Circuit
<b>IBC</b>	Intra Body Communication
<b>IBM</b>	International Business Machines
<b>IDE</b>	Integrated Development Environment
<b>IEC</b>	International Electrotechnical Commission
<b>IF</b>	Intermediate Frequency
<b>ISO</b>	International Organization for Standardization
<b>JST</b>	Japan Solderless Terminal
<b>LED</b>	Light-Emitting Diode
<b>LVA</b>	Low-Voltage-Activated
<b>MAC</b>	Media Access Control
<b>MIT</b>	Massachusetts Institute of Technology

<b>MOSFET</b>	Metal-Oxide-Semiconductor Field-Effect Transistor
<b>MSK</b>	Minimum-Shift Keying
<b>NFC</b>	Near-Field Communication
<b>NRZ</b>	Non-Return-to-Zero
<b>NTT</b>	Nippon Telegraph and Telephone
<b>OHA</b>	Open Handset Alliance
<b>OOK</b>	On-Off Keying
<b>OTG</b>	On-The-Go
<b>OTP</b>	One-Time Password
<b>P2P</b>	Peer-To-Peer
<b>PAN</b>	Personal Area Networks
<b>PC</b>	Personal Computer
<b>PCB</b>	Printed Circuit Board
<b>PIN</b>	Personal Identification Number
<b>POS</b>	Point Of Sale
<b>PPM</b>	Pulse-Position Modulation
<b>PSK</b>	Phase-Shift Keying
<b>PWM</b>	Pulse-Width Modulation
<b>QPSK</b>	Quadrature Phase-Shift Keying
<b>RF</b>	Radio Frequency
<b>RFID</b>	Radio Frequency Identification
<b>RSSI</b>	Received Signal Strength Indicator
<b>SMD</b>	Surface-Mount Device
<b>SMS</b>	Short Message Service
<b>SMT</b>	Surface-Mount Technology
<b>SQL</b>	Structured Query Language
<b>SOIC</b>	Small Outline Integrated Circuit
<b>SPI</b>	Serial Peripheral Interface
<b>SSOP</b>	Shrink Small-Outline Package
<b>TFA</b>	Two-Factor Authentication
<b>UART</b>	Universal Asynchronous Receiver/Transmitter
<b>USB</b>	Universal Serial Bus
<b>USSD</b>	Unstructured Supplementary Service Data
<b>WAP</b>	Wireless Application Protocol
<b>WBAN</b>	Wireless Body Area Networks
<b>WBSN</b>	Wireless Body Sensor Networks
<b>WPAN</b>	Wireless Personal Area Networks
<b>XLP</b>	eXtreme Low Power
<b>XML</b>	eXtensible Markup Language

# Capítulo 1

## Introdução

### 1.1 Motivação

No decorrer da última década, os telemóveis tornaram-se um elemento onnipresente, realizando muito mais funções do que aquelas que foram pensadas aquando da sua concepção. Entre elas, está o uso do telemóvel como meio de pagamento automático. Actualmente, é possível realizar esta funcionalidade se o telemóvel estiver dotado de um *transceiver* NFC (Near-Field Communication), usando-se para o efeito um conjunto de *standards* de comunicações entre o telemóvel e o terminal de pagamentos com ênfase colocado na segurança.

Para a segurança do NFC contribui o alcance extremamente curto da comunicação (aproximadamente 10 cm), muito inferior ao de outras tecnologias sem fios como WiFi, ZigBee e Bluetooth. Esta característica é também partilhada por um sistema BCC (Body Coupled Communication) que tem vindo a ser desenvolvido desde 2009 no Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro, apenas com a diferença de que o seu alcance é imposto pelo corpo humano, tornando-se perfeitamente adequado para ser utilizado em pagamentos automáticos.

Com este sistema, para efectuar a comunicação entre um *smartphone* e um terminal de pagamentos deixa de ser necessário aproximar os dois dispositivos fisicamente, permitindo por exemplo que o *smartphone* permaneça no bolso do utilizador durante o acto do pagamento, passando a ser apenas necessário aproximar o dedo, a mão ou outra parte qualquer do corpo humano ao terminal de pagamentos. Deste modo, os pagamentos passam a ser mais rápidos, mais seguros e também mais práticos e confortáveis para o utilizador.

A simplicidade e a praticabilidade deste sistema BCC, conjuntamente com a miniaturização dos dispositivos móveis e com a crescente tendência para a realização de pagamentos automáticos através destes dispositivos, justificam o estudo da sua viabilidade no paradigma dos sistemas de pagamento.

### 1.2 Objectivos

Esta dissertação de mestrado tem como objectivo principal o projecto e construção de um protótipo de um sistema de pagamento seguro por *smartphone* usando comunicação através do corpo humano. Desta forma, o corpo humano será usado como canal de transmissão entre o *smartphone* e o terminal de pagamentos.

O *smartphone* deverá estar equipado com o sistema operativo Android, o qual disponibiliza ferramentas para estabelecer a interface com unidades exteriores via USB, e deverá ser encarado como carteira digital do utilizador, pelo que processará toda a informação necessária para a realização de um pagamento automático bem-sucedido.

Pretende-se que o protocolo de autenticação do sistema de pagamento seja seguro e que uma eventual perda do *smartphone* não coloque nenhum problema de segurança futuro ao seu titular. Para autenticar o titular e reforçar a segurança da comunicação, deverá ser efectuada a leitura da impressão digital do utilizador ou a introdução do seu código pessoal.

Como objectivo secundário pretende-se melhorar o sistema BCC já implementado, nomeadamente a nível da redução do consumo de energia e da redução do tamanho da PCB.

A interface de interligação entre o *smartphone*, o sistema BCC e o terminal de pagamentos deverá ser implementada a par do *software* necessário para o seu correcto funcionamento.

Para demonstrar o funcionamento do sistema BCC e do sistema de pagamento deverão ser efectuados testes e aplicações práticas que permitam obter resultados conclusivos.

### 1.3 Metodologia

A realização desta dissertação de mestrado assenta num projecto de continuidade, introduzindo no sistema BCC já desenvolvido novas implementações e uma nova interface para comunicação com outros sistemas, nomeadamente *smartphones* Android.

É efectuado um estudo ao sistema BCC através da revisão do trabalho desenvolvido em anos anteriores, extraíndo-se as informações mais relevantes e apresentando-se em detalhe a tecnologia implementada. São depois efectuadas as melhorias e introduzidas as novas implementações no sistema BCC.

Prossegue-se ao estudo e construção da interface que permite a interligação do sistema BCC a dispositivos Android e dá-se início ao desenvolvimento do *software* necessário para a comunicação e para a realização de pagamentos.

São realizadas aplicações práticas e testes experimentais para demonstrar o funcionamento do sistema BCC e do sistema de pagamento implementado por via dos resultados obtidos.

Esta dissertação, além de ter como foco principal os objectivos mencionados anteriormente, destina-se também a servir de base para projectos futuros, pelo que se apresentam propostas futuras e possíveis melhorias a serem efectuadas ao sistema BCC e ao sistema de pagamento implementado.

### 1.4 Estrutura da dissertação

Esta dissertação de mestrado encontra-se dividida em 6 capítulos:

- **Capítulo 1 - Introdução**

Presente capítulo, no qual é feita a introdução a esta dissertação de mestrado. É apresentada a motivação para a sua realização, os seus objectivos, a metodologia seguida e a sua estrutura.

- **Capítulo 2 - Estado da arte**

Introduzem-se os conceitos relacionados com a tecnologia utilizada neste projecto e faz-se uma análise ao estado da arte deste tipo de tecnologias. É feita uma breve re-

visão aos trabalhos realizados em anos anteriores no âmbito do sistema BCC que tem vindo a ser desenvolvido na Universidade de Aveiro. É também analisado o estado da arte dos sistemas de pagamento e são referidas as vantagens e desvantagens de cada um relativamente à sua eficácia e segurança.

- **Capítulo 3 - Sistema de comunicação**

Apresenta-se em detalhe o sistema BCC implementado e o seu funcionamento. Destacam-se as implementações que foram mantidas e apresentam-se todas as novas implementações e melhorias efectuadas ao sistema.

- **Capítulo 4 - Sistema de pagamento**

De modo semelhante ao capítulo anterior, apresenta-se em detalhe o sistema de pagamento implementado e o seu funcionamento. Faz-se particular referência à interface de interligação construída entre o sistema BCC e os dispositivos Android, apresentando-se o *hardware* e *software* desenvolvidos e as implementações necessárias à interligação dos mesmos.

- **Capítulo 5 - Resultados**

Neste capítulo são apresentadas as experiências realizadas com os sistemas de comunicação e pagamento implementados neste projecto de dissertação e são analisados todos os resultados obtidos.

- **Capítulo 6 - Conclusão e trabalho futuro**

Último capítulo, no qual se resume todo o trabalho realizado e se retiram as conclusões mais relevantes. Termina-se esta dissertação de mestrado com propostas futuras e possíveis melhorias a efectuar no sistema implementado que poderão servir de base para projectos futuros.

- **Bibliografia**

Bibliografia utilizada durante a realização desta dissertação de mestrado.

- **Apêndices**

Esquemáticos eléctricos dos *transceivers* desenvolvidos para o sistema BCC e das placas IOIO-OTG utilizadas na interface Android desenvolvida para o sistema de pagamento.





## Capítulo 2

# Estado da arte

### 2.1 Sumário

Neste capítulo são introduzidos os principais conceitos da tecnologia envolvida neste projecto, é feita uma breve análise ao seu estado da arte e são ponderadas as suas vantagens e desvantagens. Segue-se uma revisão ao estado da arte do sistema BCC que tem vindo a ser desenvolvido na Universidade de Aveiro desde 2009 e que também foi incluído e melhorado neste projecto. De igual modo, é analisado o estado da arte dos sistemas de pagamento actuais, tendo em conta as vantagens e desvantagens das tecnologias utilizadas nesse âmbito.

### 2.2 Introdução aos conceitos de PAN, BAN e BCC

#### 2.2.1 Personal Area Networks (PAN) e Body Area Networks (BAN)

O conceito de PAN (Personal Area Network) foi introduzido e caracterizado por Thomas Zimmerman em 1995, depois de este ter descoberto que o corpo humano podia ser utilizado como meio condutor para transmissão de sinais eléctricos digitais, no decurso de uma investigação que estava a realizar na altura relativamente a sensores de posição em interfaces humanas no MIT (Massachusetts Institute of Technology) [1].

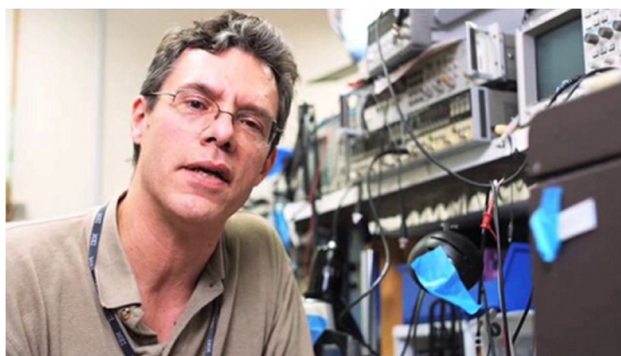


Figura 2.1. Thomas Zimmerman [2].

Zimmerman descreveu o conceito de PAN como sendo um sistema de comunicação sem fios que, através da modulação dos campos eléctricos e do acoplamento electrostático (capacitivo) de correntes da ordem dos pA no corpo humano, permite que diversos dispositivos electrónicos presentes no corpo humano ou próximos deste troquem informação digital [3].

Através do seu inovador trabalho de investigação, Zimmerman conseguiu demonstrar e comprovar este conceito [3] [4], algo que na altura foi revolucionário para a comunidade científica internacional e mesmo nos dias de hoje ainda o é para grande parte do público geral.

Nesse estudo, Zimmerman construiu um protótipo que possibilitou a comunicação através do corpo humano pelo método de acoplamento capacitivo com taxas de transmissão de aproximadamente 2.4 Kbps, utilizando uma frequência portadora de 330 KHz e as técnicas de modulação OOK (On-Off Keying) e DSSS (Direct Sequence Spread Spectrum). Apesar de a modulação DSSS ter apresentado melhor eficiência, Zimmerman decidiu optar pela modulação OOK para a demonstração do protótipo, devido à sua maior simplicidade de implementação. A figura 2.2 apresenta o protótipo construído por Thomas Zimmerman.



Figura 2.2. Protótipo desenvolvido por Thomas Zimmerman [3].

Na sua demonstração, Thomas Zimmerman utilizou um transmissor contendo um microcontrolador que enviava continuamente caracteres ASCII representando um cartão electrónico e um receptor para receber a informação. O transmissor encontrava-se próximo do pé de uma pessoa e o receptor próximo do pé de outra pessoa, ambos simulando dispositivos inseridos nos seus respectivos sapatos. Através de um simples aperto de mão, o circuito ficaria completo, permitindo que uma corrente da ordem dos pA atravessasse os corpos dos dois indivíduos desde o dispositivo transmissor até ao dispositivo receptor e efectuando desta forma a transferência da informação. O sinal recebido era depois desmodulado e enviado para um computador através de uma ligação série para se visualizarem os caracteres recebidos, conforme ilustra a figura 2.3.

Devido ao sucesso da sua implementação (desenvolvida no âmbito da sua dissertação de mestrado), Thomas Zimmerman acabou por despertar o interesse e por impulsionar a investigação nesta área a nível internacional. Em pequena nota de curiosidade, Zimmerman trabalha actualmente como investigador no Almaden Research Center da IBM em San José, Califórnia, EUA.

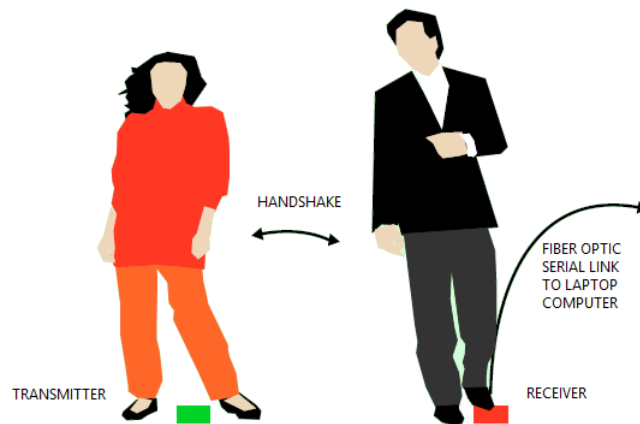


Figura 2.3. Experiência realizada por Thomas Zimmerman [4].

O conceito de PAN foi evoluindo com o decorrer do tempo e, por volta de 2001, acabou por se tornar mais generalista, mais abrangente e mais adequado à sua nomenclatura, isto é, abstraiu-se da sua camada física [5]. Assim, uma PAN será a rede de dispositivos pessoais do utilizador na qual os dispositivos comunicam entre si independentemente da sua localização e dos meios de transmissão que usam para o efeito. Esses meios poderão ser, por exemplo, tecnologias RF ou a Internet (via *uplink*). Por esta razão, também é comum a designação das PAN por WPAN (Wireless Personal Area Networks). Na figura 2.4, ilustra-se um exemplo de uma PAN.



Figura 2.4. Exemplo de uma PAN.

Foi então nesta linha de seguimento que surgiu o conceito de BAN (Body Area Network), sendo este caracterizado por uma rede de dispositivos e sensores que comunicam e operam sobre o corpo humano ou nas imediações deste [3]. Também para as BAN é comum a designação de WBAN (Wireless Body Area Networks), WBSN (Wireless Body Sensor Networks) ou HAN (Human Area Networks).

Portanto, uma BAN será um subdomínio de uma PAN mas com a particularidade de ser uma rede onde os dispositivos apenas comunicam sobre o corpo humano ou nas suas imediações, tipicamente através de tecnologias RF como WiFi, Bluetooth, ZigBee, RFID, NFC ou então através de tecnologias BCC como se verá mais adiante neste capítulo. Na figura 2.5, ilustra-se um exemplo de uma BAN.

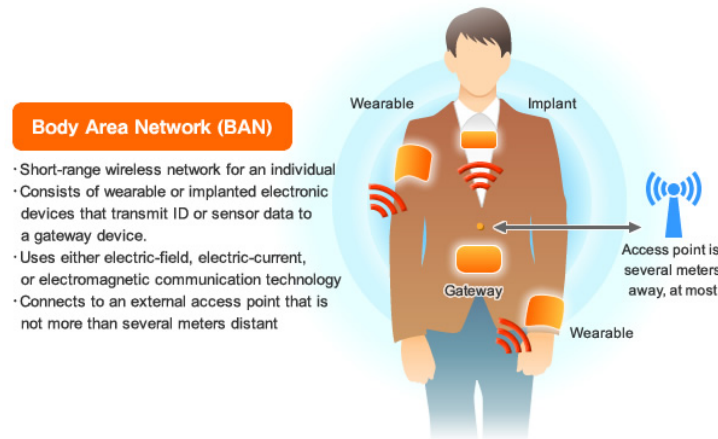


Figura 2.5. Exemplo de uma BAN [6].

Na figura 2.6 podem visualizar-se exemplos práticos de interação entre uma BAN e uma PAN no mundo real e, na figura 2.7, um diagrama que ilustra o posicionamento de uma PAN e uma BAN no paradigma das redes sem fios (*wireless*).

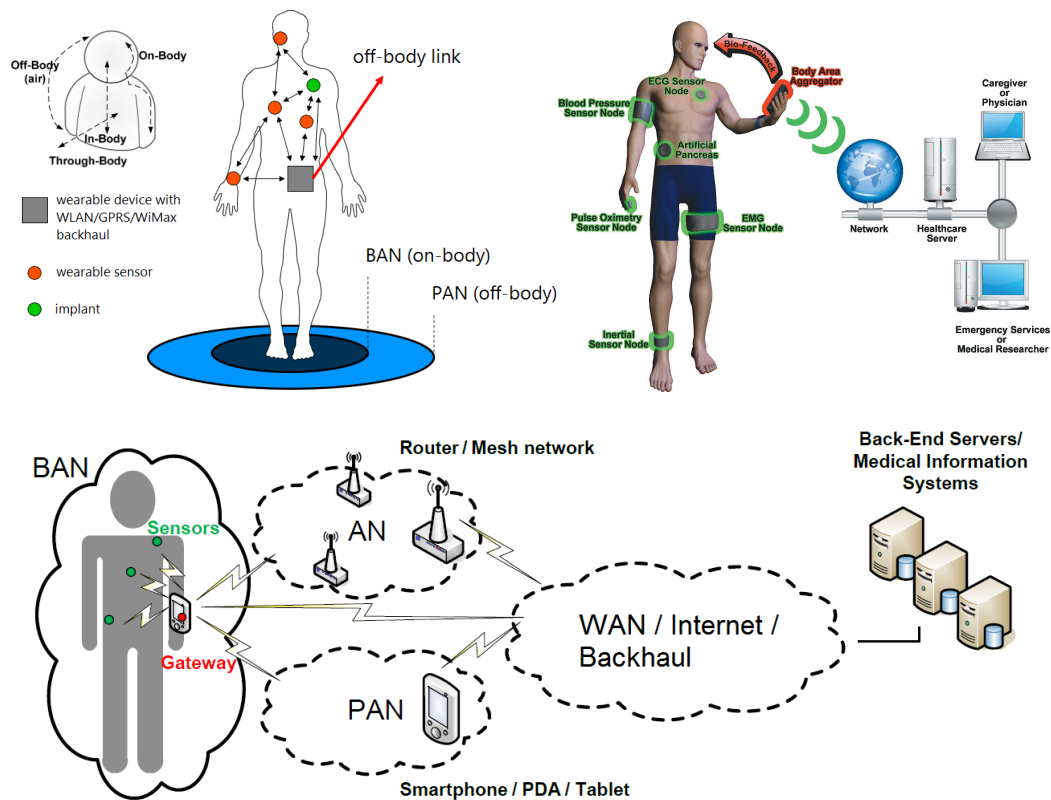


Figura 2.6. Exemplos práticos de interação entre PAN e BAN [7] [8].

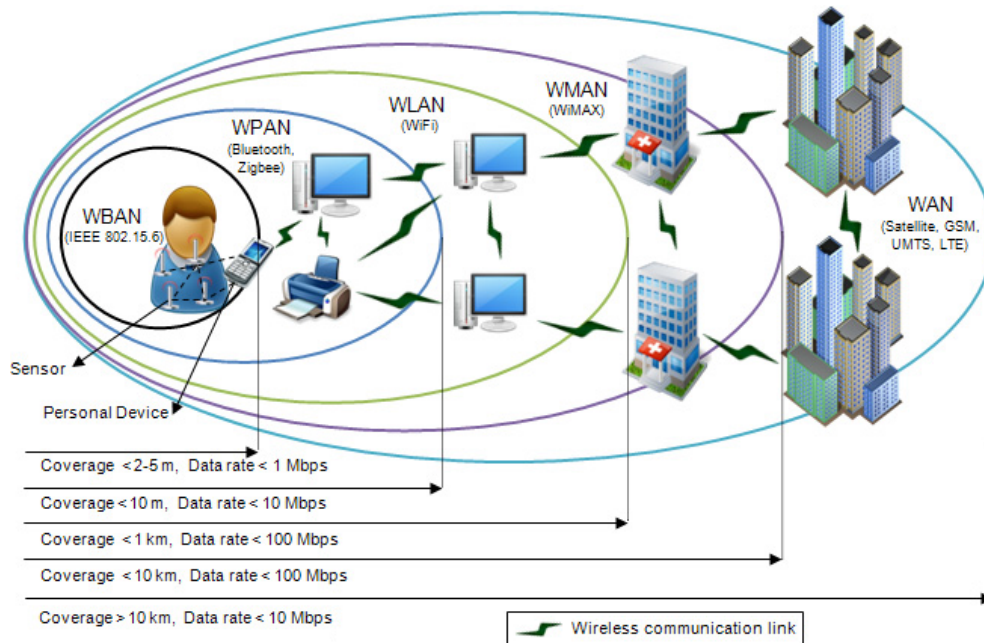


Figura 2.7. Posicionamento de uma PAN e de uma BAN no paradigma das redes sem fios (*wireless*) [9].

Contudo, as tecnologias presentes nestes conceitos ainda apresentam diversas limitações, entre elas, o elevado consumo de energia, a significativa contaminação do espectro (que se traduz em possíveis interferências noutros dispositivos) e, no caso particular das BAN, um alcance demasiado elevado que por sua vez compromete tanto o consumo de energia dos dispositivos como a segurança da informação trocada entre estes. Na figura 2.8 é possível observar algumas das limitações presentes nas tecnologias utilizadas nas PAN e BAN actuais comparativamente às características de uma BAN ideal.

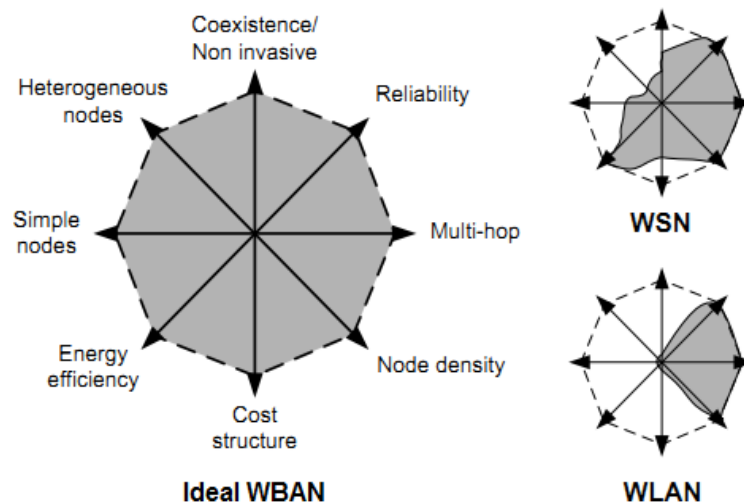


Figura 2.8. Algumas limitações das tecnologias utilizadas nas PAN e BAN actuais comparativamente às características de uma BAN ideal [10].

### 2.2.2 Body Coupled Communication (BCC)

Apesar de ter sido inicialmente descrito como PAN, o conceito de BCC (Body Coupled Communication) foi na realidade o conceito que surgiu com o trabalho de Thomas Zimmerman em 1995. Outras designações comuns para BCC são as de HBC (Human Body Communication) e IBC (Intra Body Communication).

Como já foi referido, a tecnologia BCC utiliza o corpo humano como meio de propagação para a transmissão de informação digital. Deste modo, existem duas abordagens possíveis para injectar um sinal digital no corpo humano: ou por acoplamento galvânico ou por acoplamento capacitivo. No acoplamento galvânico, o transmissor e o receptor têm de estar obrigatoriamente em contacto directo com o corpo humano, o qual é utilizado como linha de transmissão (ou guia de onda). No acoplamento capacitivo, tanto o transmissor como o receptor não necessitam de estar em contacto directo com o corpo humano, mas sim, próximos deste, e o potencial eléctrico (campo eléctrico) em torno do corpo é modificado em função do sinal a ser transmitido [11]. Estas abordagens irão ser estudadas em mais pormenor na secção 2.3.

Depois de um período de estagnação desta tecnologia, entre 2003 e 2007 a mesma voltou a despertar atenções e a ganhar novamente interesse no âmbito da investigação internacional. Alguns dos maiores avanços feitos nesta área partiram da contribuição da NTT (Nippon Telegraph and Telephone) em 2004. Durante este período, realizaram-se diversas experiências onde diferentes técnicas foram utilizadas de modo a serem obtidas elevadas taxas de transmissão de dados. Esta meta foi atingida com maior sucesso através de implementações opto-eléctricas por acoplamento capacitivo, chegando a obterem-se taxas de transmissão de aproximadamente 10 Mbps [1] [12]. Uma das maiores contribuições neste âmbito foi fruto do trabalho de Mitsuru Shinagawa, investigador da NTT Microsystems Integration Laboratories.

Na tabela 2.1, apresentam-se algumas implementações importantes em BCC que foram realizadas nos últimos anos por diversos investigadores a nível mundial. Entre elas encontram-se a contribuição de Thomas Zimmerman e a de Mitsuru Shinagawa.

Um dos motivos pelo qual o BCC tem sido alvo de maior investigação e desenvolvimento nos últimos anos provém do facto de ser uma solução promissora para os problemas apresentados pelas tecnologias utilizadas nas BAN actuais. O BCC proporciona maior segurança, maior eficiência no consumo de energia e maior conforto para o utilizador num elevado número de aplicações. Além disso, não apresenta interferência entre dispositivos utilizados por diferentes pessoas que se encontrem na mesma proximidade. Todas estas vantagens devem-se ao facto de a comunicação estar confinada apenas ao corpo humano e de o seu alcance ser limitado por este.

Nas figuras 2.9 e 2.10, relativamente ao consumo de energia, taxa de transmissão e distância de propagação no ar, são comparadas várias tecnologias de comunicação sem fios (*wireless*) utilizadas vulgarmente nas PAN e BAN actuais. Neste âmbito, é de realçar que o posicionamento ideal de uma BAN (menor consumo de energia, maior taxa de transmissão e menor distância de propagação no ar) pode ser alcançado com a introdução de tecnologias BCC, como se pode visualizar.

	<b>Zimmerman</b>	<b>Shinagawa</b>	<b>Hachisuka</b>
<b>Ano(s)</b>	1995	2003-2004	2003-2005
<b>Modelo</b>	Eléctrico	-	-
<b>Acoplamento</b>	Capacitivo	Capacitivo	Capacitivo
<b>Largura de banda</b>	100 KHz - 500 KHz	-	10 KHz - 50 GHz
<b>Modulação</b>	OOK	-	FSK
<b>Taxa de transm.</b>	2.4 Kbps	10 Mbps	9.6 Kbps
<b>Localização no corpo humano</b>	Braço, Pé, Cintura, Cabeça	Mão	Braço
<b>Tamanho do eléctrodo</b>	25x25 mm <sup>2</sup> 30x30 mm <sup>2</sup> 70x20 mm <sup>2</sup> 80x80 mm <sup>2</sup> 130x40 mm <sup>2</sup> 900x25 mm <sup>2</sup>	-	30x30 mm <sup>2</sup> 20x20 mm <sup>2</sup>
<b>Orientação</b>	Vertical	Eléctrodo singular	Horizontal
<b>Tipo do receptor</b>	Ampl. de corrente seguido por <i>chopper</i>	Sensor opto-eléctrico de luz laser	Receptor FM intracorporal

	<b>Song</b>	<b>Wegmüller</b>	<b>Lin</b>
<b>Ano(s)</b>	2006-2007	2007	2011
<b>Modelo</b>	Eléctrico	Eléctrico	-
<b>Acoplamento</b>	Capacitivo	Galvânico	Ag/AgCl
<b>Largura de banda</b>	1 MHz - 200 MHz	1 KHz - 10 MHz	50 KHz - 300 KHz
<b>Modulação</b>	PPM	FSK, BPSK	OOK
<b>Taxa de transm.</b>	10 Mbps	128, 255 Kbps	2 Mbps
<b>Localização no corpo humano</b>	-	Braço, Pulso, Perna	Pulso
<b>Tamanho do eléctrodo</b>	-	880 mm <sup>2</sup> 54 cm <sup>2</sup> 560 cm <sup>2</sup>	-
<b>Orientação</b>	-	Horizontal	-
<b>Tipo do receptor</b>	Amplificador seguido por um Schmitt <i>trigger</i>	-	Receptor com LVA, ampl. em cascata, ampl. STD, LVM, filtro passa-baixo, comparador e buffer

Tabela 2.1. Implementações BCC realizadas nos últimos anos por vários investigadores de todo o mundo [12].

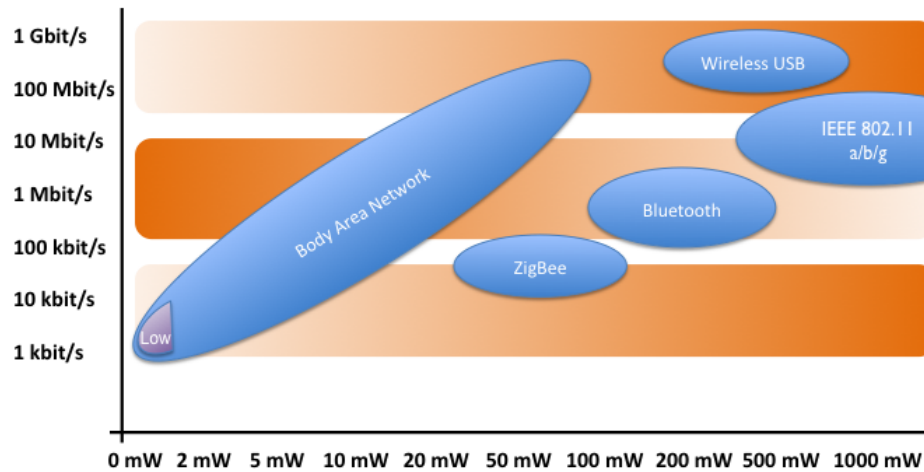


Figura 2.9. Gráfico comparativo das várias tecnologias sem fios (*wireless*) utilizadas nas PAN e BAN actuais (taxa de transmissão vs potência consumida) [13].

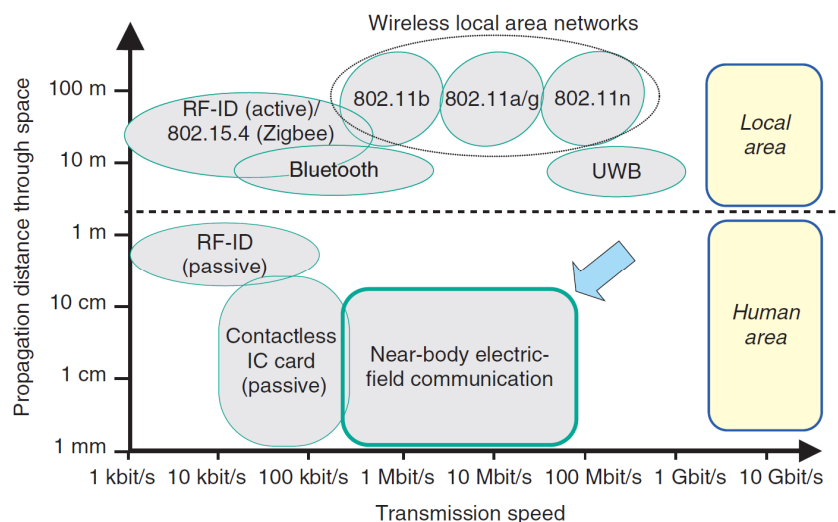


Figura 2.10. Gráfico comparativo das várias tecnologias sem fios (*wireless*) utilizadas nas PAN e BAN actuais (distância de propagação no ar vs taxa de transmissão) [14].

Tornando-se possível a existência de um cenário onde o corpo humano é usado como canal de transmissão de um sistema de comunicação, surge um infindável leque de aplicações, como por exemplo:

- Trocar de dados pessoais através de um simples aperto de mão.
- Ouvir música transmitindo a informação através corpo humano desde o dispositivo até aos auriculares sem fios.
- Obter informações ou efectuar pagamentos através do toque ou por aproximação a um terminal.



- Monitorizar pacientes de um hospital sem a necessidade do habitual e incómodo aparato de cabos.
- Abrir a porta de um carro ou de uma casa apenas com um toque na maçaneta ou simplesmente por aproximação.
- Muitas outras...

As figuras 2.11 e 2.12 ilustram alguns exemplos de aplicações que podem fazer uso de BCC e as novas capacidades das BAN com a introdução destas tecnologias, respectivamente.

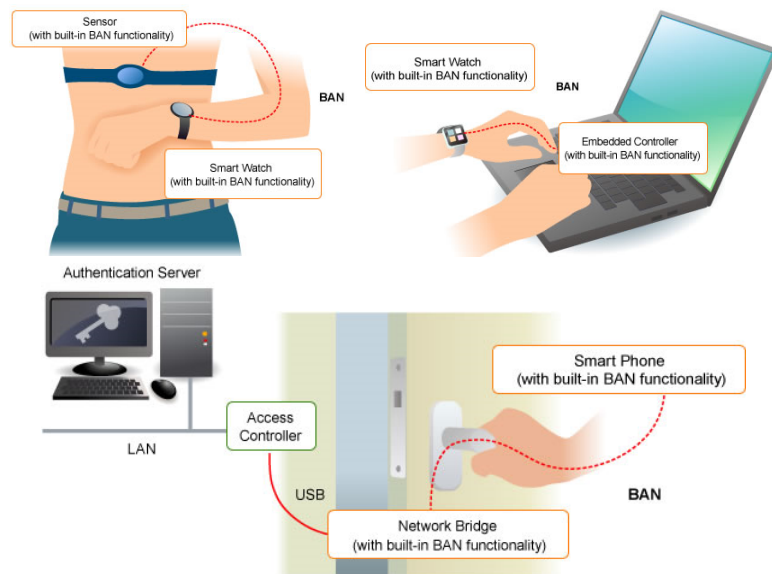


Figura 2.11. Exemplos de aplicações que podem utilizar BCC [6].



Figura 2.12. Novas capacidades das BAN com a introdução de tecnologias BCC [15].

O BCC mesmo tendo actualmente as suas próprias limitações como qualquer outra tecnologia, tem ainda bastantes margens para evoluir e para vir a ser uma excelente solução para todas estas aplicações. Consequentemente, poderá abrir portas a um mercado bastante atractivo, não só para a electrónica de consumo mas também para diversos sectores como, por exemplo, os sectores médico, militar e industrial.

## 2.3 Body Coupled Communication (BCC)

### 2.3.1 Introdução

O BCC (Body Coupled Communication) é uma técnica que utiliza o corpo humano como canal de transmissão para comunicações sem fios (*wireless*) através da modulação dos campos eléctricos que se encontram nas suas imediações.

Esta técnica distingue-se de muitas outras precisamente pelo facto de não radiar ondas electromagnéticas na comunicação. Ao invés, utiliza *transceivers* que geram fracos (no entanto detectáveis) campos eléctricos que se acoplam ao corpo humano, permitindo a transmissão de informação desde o emissor até ao receptor [11]. Devido ao facto de usar o corpo humano como canal de comunicação, não sofre de interferências e possibilita transmissões de dados eficientes e seguras entre o emissor e o receptor. Outras das suas maiores vantagens são o seu baixo consumo, as reduzidas perdas de propagação (abaixo de 80 dB), a baixa atenuação na situação de movimento do corpo e a eficiência dos *transceivers* habitualmente utilizados [16] [17] [18]. A tabela 2.2 compara de modo qualitativo o estado actual da tecnologia BCC relativamente a outros tipos de comunicação.

	Com fios	Sem fios	Corpo humano
Portabilidade	Baixa	Alta	Alta
Consumo de energia	Alto	Alto	Baixo / Alto
Taxa de transmissão	Alta	Baixa / Alta	Baixa / Alta
Perdas de propagação	Baixas	Altas	Baixas
Interferência	Baixa	Alta	Baixa
Segurança	Alta	Baixa	Alta

Tabela 2.2. Comparação qualitativa da tecnologia BCC relativamente a outros tipos de comunicação. Baseado em [18].

### 2.3.2 Abordagens em BCC

Como já foi referido, existem duas abordagens possíveis para a implementação de tecnologias BCC. Elas são:

- Acoplamento galvânico
- Acoplamento capacitivo

Na abordagem por acoplamento galvânico, o corpo humano é utilizado como linha de transmissão (ou guia de onda) e são utilizados eléctrodos que necessitam de estar em contacto directo com este para lhe serem aplicados os sinais a transmitir. Um par de eléctrodos pertence ao emissor e outro par de eléctrodos pertence ao receptor. Ambos são acoplados ao corpo, pelo que não necessitam de uma referência externa (*ground*).

Para transmitir um sinal, primeiro este é aplicado diferencialmente aos eléctrodos do emissor. Um fluxo de corrente (primário) é assim estabelecido entre o par de eléctrodos do emissor e consequentemente outro fluxo de corrente (secundário) é induzido e propagado pelo corpo humano até ao par de eléctrodos do receptor. A indução deste fluxo de corrente secundário resulta numa diferença de potencial entre os eléctrodos do receptor, permitindo que este detecte o sinal transmitido pelo emissor [19].

Na figura 2.13, apresenta-se o conceito de acoplamento galvânico aplicado ao corpo humano a par do seu respectivo modelo de representação. Em última nota, refere-se que até aos dias de hoje foram desenvolvidos poucos protótipos que façam uso de acoplamento galvânico [19]. Em 2007, Marc Wegmüller foi um dos investigadores que desenvolveu um protótipo deste tipo, quando estudava a possibilidade de modelar o corpo humano como canal de comunicação [20].

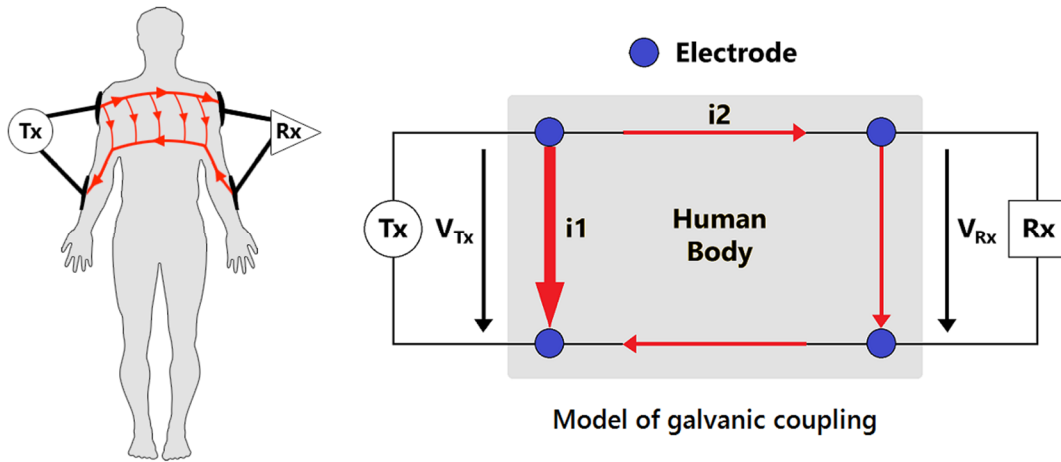


Figura 2.13. Acoplamento galvânico. Baseado em [19] [21].

Na abordagem por acoplamento capacitivo, os campos eléctricos nas imediações do corpo humano são modulados por pares de eléctrodos usados no emissor e no receptor. Os eléctrodos de ambos não necessitam de estar em contacto directo com o corpo, bastando apenas que se encontrem próximos deste. Contudo, necessitam de uma referência externa (*ground*).

Para transmitir um sinal, primeiro este é aplicado diferencialmente aos eléctrodos do emissor. Como os eléctrodos têm diferentes acoplamentos ao corpo, é induzido um campo eléctrico neste que o atravessa e que varia o seu potencial eléctrico. De igual modo, os eléctrodos do receptor também têm diferentes acoplamentos ao corpo e, como tal, a variação do campo eléctrico induzida pelo emissor resulta numa diferença de potencial entre os eléctrodos do receptor, permitindo assim que este detecte o sinal diferencial transmitido. Devido ao facto de este ser um sinal não radiante, tem de ser fornecido um caminho de retorno (referência), pelo que se utiliza o ambiente externo para esse efeito.

Desta forma, o corpo humano é modelado como um condutor perfeito que faz a ponte entre o emissor e o receptor. Por sua vez, os acoplamentos eléctricos entre os eléctrodos, o corpo

humano e o ambiente externo são modelados por capacidades [16] [22]. A figura 2.14 apresenta o conceito de acoplamento capacitivo aplicado ao corpo humano.

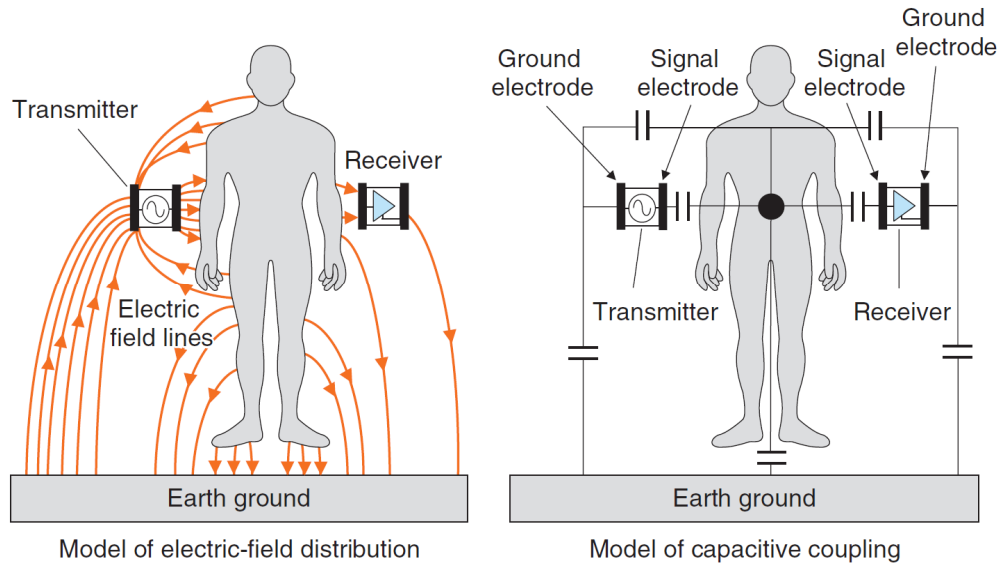


Figura 2.14. Acoplamento capacitivo [14].

Ambas as abordagens apresentadas têm as suas vantagens e desvantagens. Na abordagem por acoplamento galvânico os eléctrodos têm de estar em contacto directo com o corpo humano e apenas podem ter estrutura horizontal. Por sua vez, na abordagem por acoplamento capacitivo basta encontrarem-se próximos do corpo e podem ter estrutura tanto horizontal como vertical. A figura 2.15 apresenta as estruturas horizontal e vertical dos eléctrodos.

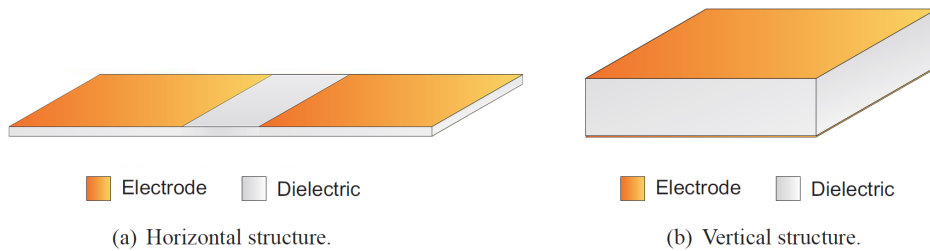


Figura 2.15. Estruturas horizontal e vertical dos eléctrodos [16].

Note-se que para a estrutura horizontal, os eléctrodos podem estar orientados tanto longitudinalmente como transversalmente ao corpo humano, como se pode visualizar na figura 2.16.

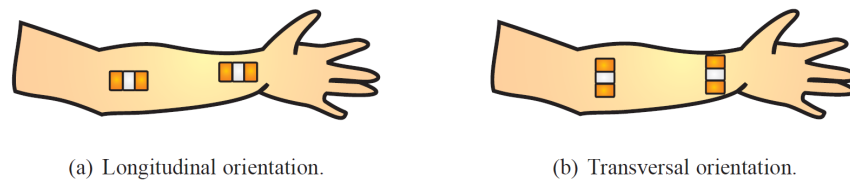


Figura 2.16. Orientações para a estrutura horizontal dos eléctrodos [16].

A técnica de acoplamento capacitivo foi a abordagem escolhida para o sistema BCC implementado neste projecto. Irá ser feita uma breve introdução ao sistema na secção 2.4 e, no capítulo 3, este será estudado em mais pormenor.

### 2.3.3 Alcance e distribuição do campo eléctrico

Relativamente ao campo eletromagnético ou à radiação electromagnética de um objecto (uma antena, por exemplo), existem duas regiões: *near-field* e *far-field*.

A região *near-field* corresponde aos campos electromagnéticos não-radiantes, os quais têm maior predominância nas proximidades da antena ou do objecto. Por sua vez, a região *far-field* corresponde à radiação electromagnética com predominância em distâncias maiores. Na região *near-field*, a amplitude do campo eléctrico decai mais rapidamente com a distância ( $r$ ) ao objecto (decaimento proporcional a  $r^{-3}$ ), enquanto que na região *far-field*, a amplitude decai mais lentamente (decaimento proporcional a  $r^{-1}$ ) [23] [24].

Outra diferença fundamental entre as regiões *near-field* e *far-field* consiste no projecto da antena. Na região *near-field*, o acoplamento electrostático é proporcional à área da superfície do eléctrodo. Na região *far-field*, a eficiência de transmissão é maximizada pela adaptação da impedância do transmissor à impedância do ar (ou espaço livre), onde são utilizadas tipicamente antenas com meio comprimento de onda [4].

As figuras 2.17 e 2.18 apresentam as regiões *near-field* e *far-field* e o seu modelo de três regiões, respectivamente.

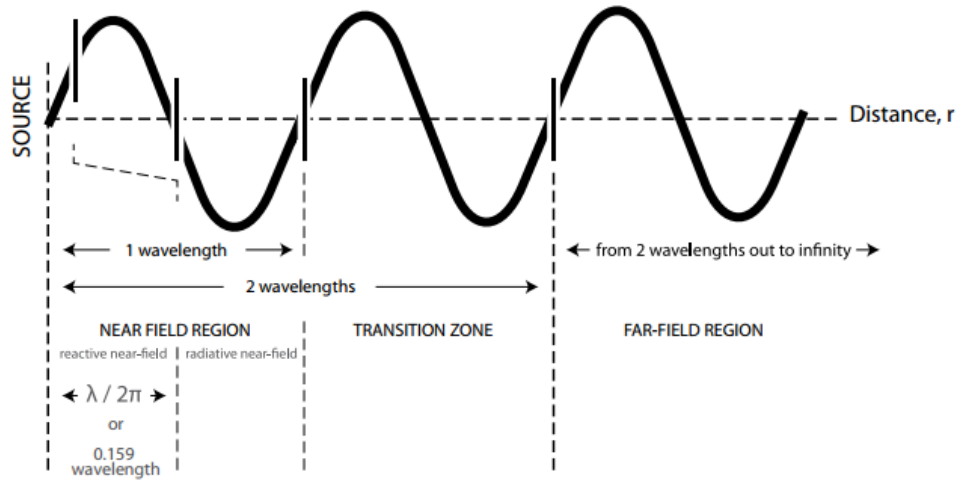


Figura 2.17. Regiões *near-field* e *far-field* [23].

THREE-REGION MODEL			
DOMINANT TERMS IN THE REGION	$\frac{1}{r}$	$\frac{1}{r^2}$	$\frac{1}{r^3}$
	FAR FIELD	TRANSITION ZONE	NEAR FIELD
	FRAUNHOFER ZONE	INDUCTION FIELD	FREEL ZONE
	FAR FIELD	INDUCTION FIELD	STATIC FIELD
	FAR RADIATION FIELD	NEAR RADIATION FIELD	REACTIVE FIELD
	FAR FIELD	TRANSITION REGION	QUASISTATIONARY REGION

Figura 2.18. Modelo de três regiões *near-field* e *far-field* [24].

O alcance de transmissão de um sistema BCC é por norma reduzido pelo facto de se encontrar confinado ao corpo humano ou às suas imediações, permitindo desta forma menor interferência, maior segurança e também maior eficiência energética. A sua região é, portanto, a *near-field*. Ao contrário das tecnologias RF onde toda a energia é radiada para o ar independentemente de existirem ou não receptores, na tecnologia BCC apenas uma pequena parte da energia é perdida entre o emissor e o receptor, devido à natureza reactiva da sua região (*near-field*).

Em [25] foi efectuado um estudo sobre a distribuição do campo eléctrico no corpo humano onde se investigou qual a sua dependência com a frequência portadora. Efectuaram-se várias experiências onde se transmitiram sinais através do corpo humano e posteriormente se analisaram as distribuições do campo eléctrico para as frequências de 3 MHz, 30 MHz, 300 MHz e 3 GHz. A potência de transmissão foi de 1 mW. Os resultados obtidos são apresentados na figura 2.19 (note-se nos gráficos que 0 dB corresponde a 500 V/m) e a eficiência de radiação para os diversos casos é apresentada na tabela 2.3.

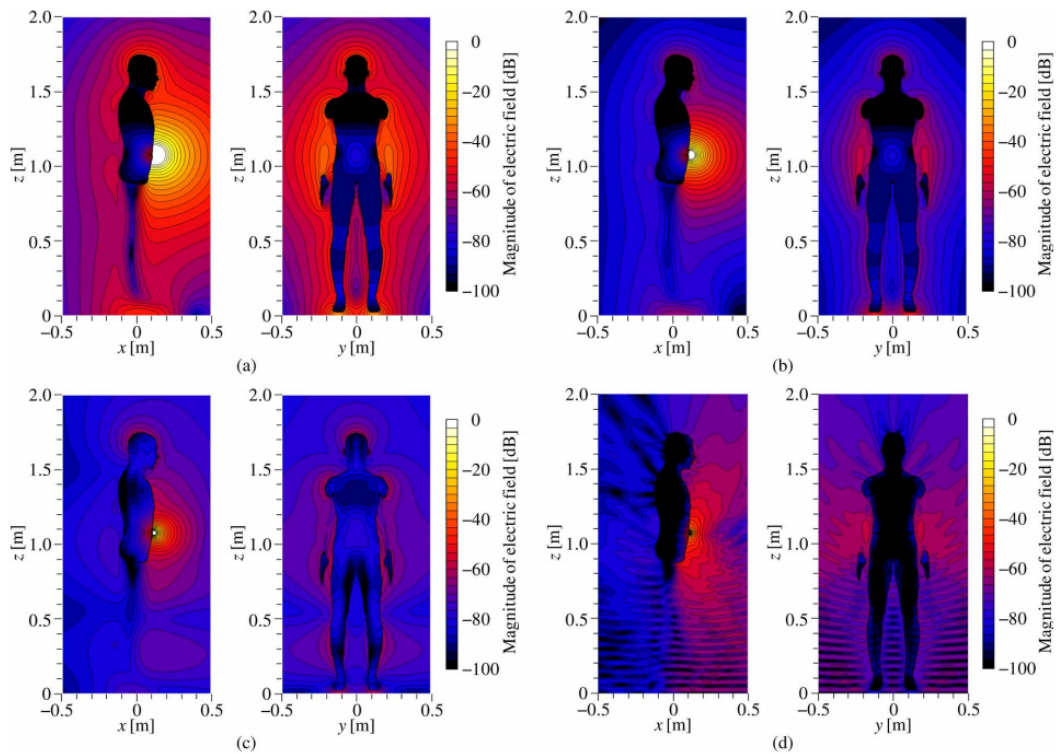


Figura 2.19. Distribuições do campo eléctrico para várias frequências. (a) 3 MHz. (b) 30 MHz. (c) 300 MHz. (d) 3 GHz. [25].

Frequência	Eficiência de radiação
3 MHz	0.0014 %
30 MHz	0.081 %
300 MHz	6 %
3 GHz	49 %

Tabela 2.3. Eficiência de radiação [25].

Pela análise dos resultados obtidos verifica-se que a 3 MHz e 30 MHz as distribuições são semelhantes, apesar de a 30 MHz a magnitude ser 20 dB menor. A 300 MHz, a magnitude reduz-se ainda mais mas a distribuição continua a ter semelhanças relativamente às das frequências anteriores. A 3 GHz já se verifica uma distribuição bastante diferente e inclusive *ripple* acentuado. Isto deve-se à redução do comprimento de onda para altas frequências que, por sua vez, provocam variações bruscas no campo eléctrico. Estas variações induzem radiação electromagnética para fora do corpo humano e este acaba por operar como uma antena radiando o sinal para o ar.

Como tal, conclui-se que a distribuição do campo eléctrico depende da frequência portadora e que esta limita consideravelmente a implementação de um sistema BCC. Assim, a frequência deve ser escolhida de modo a que o comprimento de onda seja suficientemente maior que o corpo humano, permitindo que o campo eléctrico gerado pelo emissor cubra a totalidade do corpo e reduza a eficiência de radiação.

O estudo realizado em [26] vem também confirmar estas conclusões, pelo que na figura 2.20 se apresentam mais alguns resultados interessantes obtidos no mesmo âmbito, fruto desta investigação.

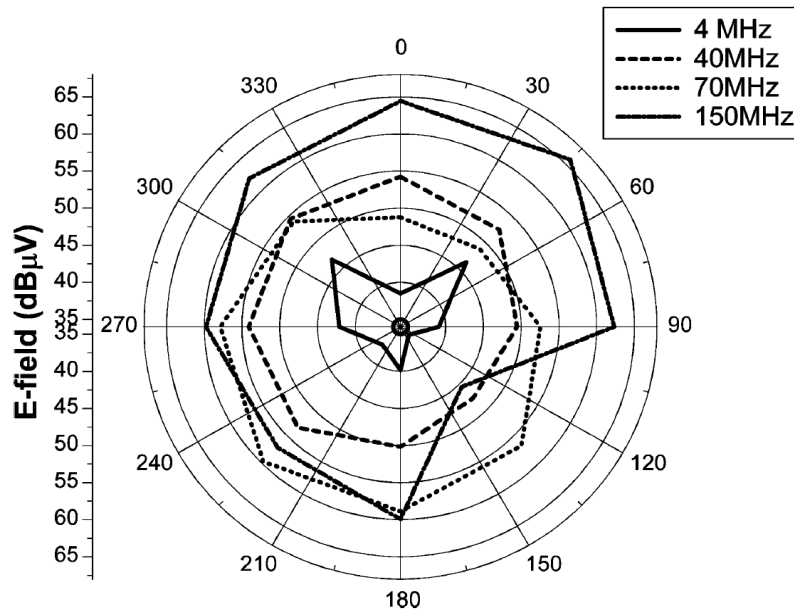


Figura 2.20. Distribuição do campo eléctrico para diferentes frequências [26].

Além das conclusões anteriormente mencionadas, neste estudo observou-se que as diferentes morfologias e posições do corpo humano afectam de forma diferente as medições, notando-se para a mesma frequência uma maior radiação na presença de indivíduos mais altos, algo que pode ser explicado pelo facto de estes possuírem frequências de ressonância mais baixas e mais próximas da região *near-field* [26].

Em [27] procurou-se comparar a atenuação sofrida por um sinal ECG no caso em que este era enviado através do corpo humano e no caso em que era radiado pelo ar. Para tal utilizaram-se dois *transceivers* (um emissor e um receptor) e eléctrodos circulares com dimensões entre 1 e 3 cm. As modulações usadas foram OOK e FSK com frequências portadoras de 400 MHz e 480 MHz e efectuaram-se medições do ganho do sinal recebido até uma distância máxima de 50 cm. Note-se que para o caso em que o canal de transmissão era o ar, os eléctrodos simples-

mente foram deixados a flutuar, isto é, não foram acoplados ao corpo humano ou a qualquer outro meio de propagação. Na figura 2.21, apresentam-se os resultados obtidos neste estudo.

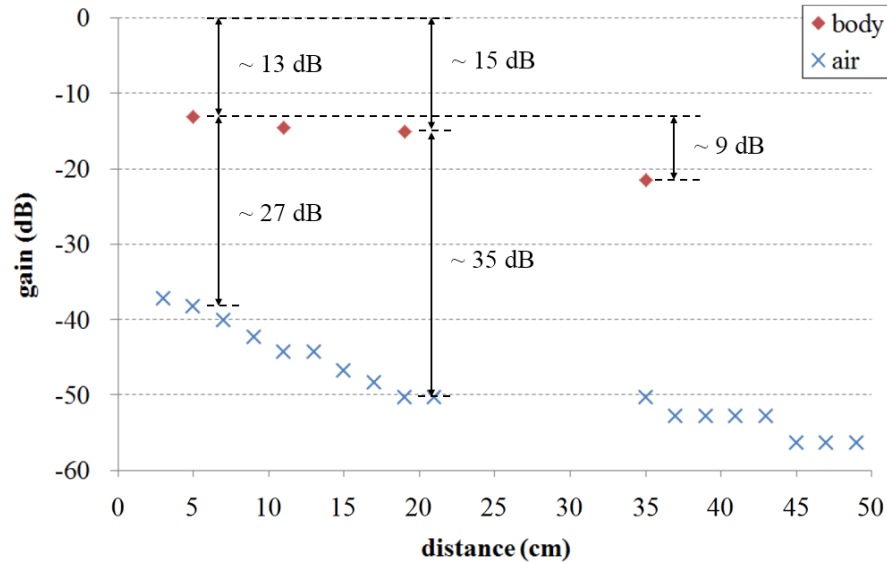


Figura 2.21. Ganho do sinal recebido para os dois casos [27].

A partir do gráfico é possível listar (de grosso modo) os seguintes valores:

Distância	Atenuação (ar)	Atenuação (corpo)
5 cm	~ 40 dB	~ 13 dB
20 cm	~ 50 dB	~ 15 dB
35 cm	~ 50 dB	~ 22 dB
50 cm	~ 57 dB	-

Tabela 2.4. Atenuação do sinal recebido para as diferentes distâncias.

Como se pode observar pela figura 2.21 e pela tabela 2.4, a atenuação do sinal recebido é bastante maior no caso em que o canal de transmissão é o ar. Além disso, verifica-se que neste caso o sinal sofre considerável atenuação logo nos primeiros 5 cm de distância entre os electrodos. Estes resultados vêm assim comprovar o curto alcance da tecnologia BCC quando o meio de propagação é o ar, garantindo por isso maior segurança e reduzida interferência, como já se referiu anteriormente.

Como nota final, conclui-se que a comunicação em BCC está praticamente confinada ao corpo humano desde que durante a implementação do sistema BCC se escolha correctamente a frequência portadora.

### 2.3.4 Frequência portadora ideal

Diferentes canais de transmissão possuem diferentes características. Essas características são normalmente os factores que influenciam fortemente o comportamento de um canal de transmissão e que determinam a que frequências é que um sinal é mais (ou menos) atenuado.



Dado que em BCC o corpo humano é modelado como canal de transmissão, torna-se crucial a análise ao seu comportamento para diferentes frequências portadoras numa transmissão de dados.

Em [28] realizou-se um estudo que tinha como um dos objectivos a determinação da frequência portadora ideal para a transmissão de dados através do corpo humano.

Foram realizadas diversas experiências entre as quais se comparou o corpo humano com o ar relativamente ao seu uso como canal de transmissão. A figura 2.22 mostra a configuração utilizada nas experiências realizadas.

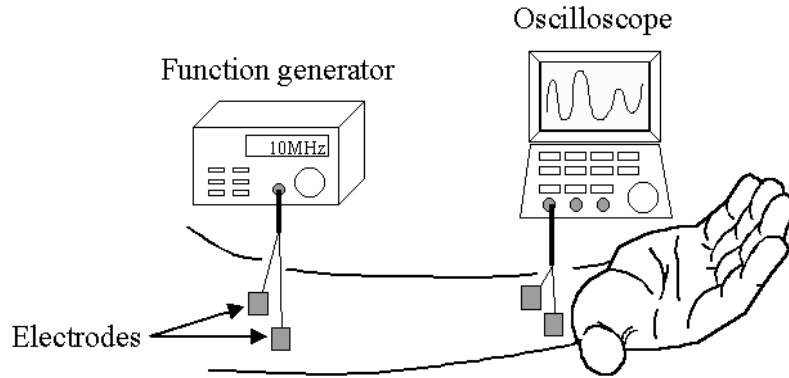


Figura 2.22. Configuração utilizada nas experiências realizadas em [28].

O gerador de funções gerava ondas sinusoidais entre 1 e 40 MHz com amplitude de 1 Vpp que depois eram aplicadas ao corpo através de eléctrodos e se propagavam neste até alcançarem os eléctrodos do osciloscópio, o qual por sua vez media o sinal recebido. Nas figuras 2.23 e 2.24, são apresentados os resultados obtidos.

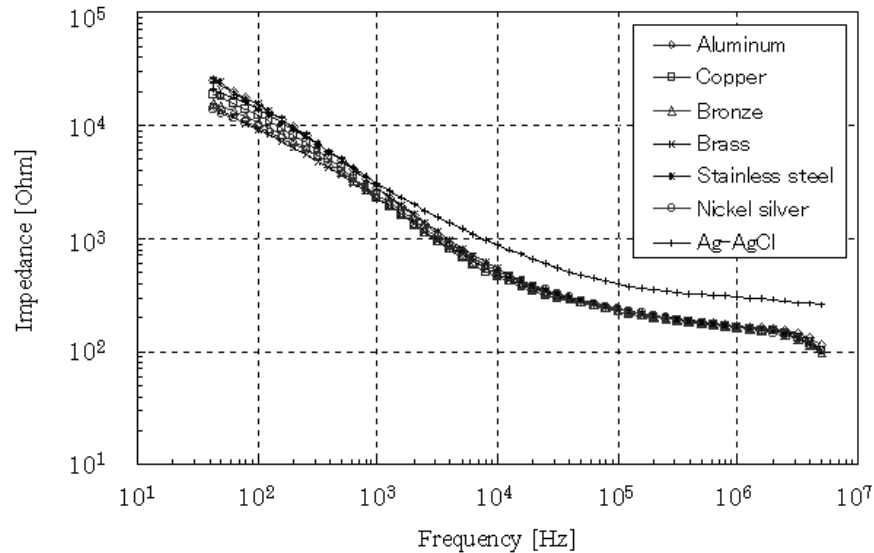


Figura 2.23. Comparação entre as impedâncias obtidas no contacto do corpo humano com eléctrodos de diferentes metais [28].

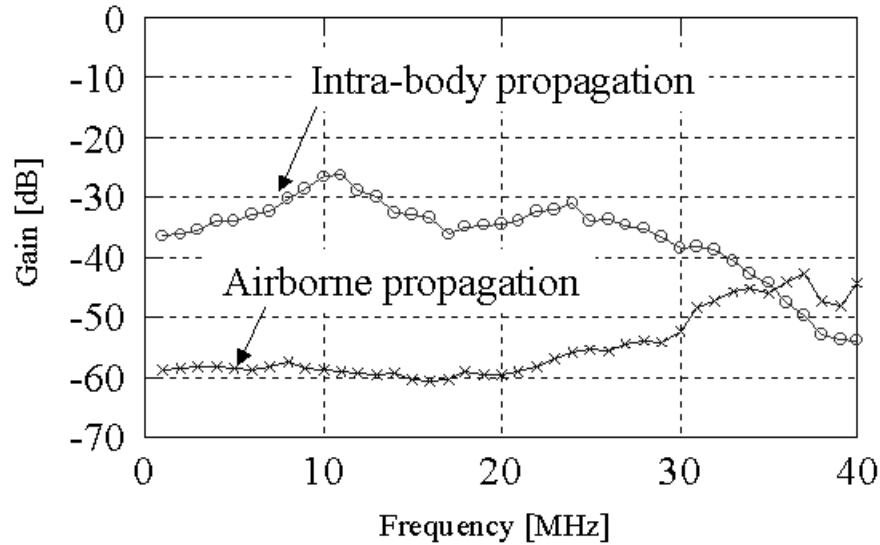


Figura 2.24. Comparação entre os ganhos obtidos para diferentes frequências portadoras utilizando o corpo humano e o ar como canais de transmissão [28].

Como se pode observar pelas figuras 2.23 e 2.24, o corpo humano apresentou, respectivamente, a menor impedância e o maior ganho para uma frequência portadora de aproximadamente 10.7 MHz. Assim, segundo este estudo, esta será a frequência portadora ideal para a utilização do corpo humano como canal de transmissão.

### 2.3.5 Eficiência da transmissão

Em [29] foi realizado um estudo relativamente ao uso do corpo humano como canal de comunicação, no qual se efectuaram diversas experiências com o objectivo de testar a sua eficiência perante diferentes taxas de transmissão de dados e diferentes distâncias entre o emissor e o receptor. Foram utilizadas as modulações MSK, BPSK e QPSK com frequências portadoras de 200, 300, 400 e 500 MHz e taxas de transmissão de 100 Kbps e 2.5 Mbps. As distâncias de teste entre o emissor e o receptor foram de 20 cm e 155 cm. Foram efectuadas medições para o caso em que o canal de transmissão é o corpo humano e outras para o caso em que o canal de transmissão é o ar. A figura 2.25 mostra as constelações obtidas para a frequência portadora de 200 MHz, dado que para as outras frequências os resultados foram semelhantes.

Como se pode observar pelas constelações obtidas, neste estudo concluiu-se que o corpo humano é um canal de transmissão fiável, garantindo melhor performance e menor distorção do que nos casos em que a transmissão é feita via ar. Isto deve-se ao facto de o corpo se tornar uma espécie de fio condutor que não é afectado por interferências externas ou fontes de ruído presentes no ar [29]. O estudo realizado em [28] confirma igualmente esta característica.

Como nota final, concluiu-se também que a distorção aumenta com a distância e com a taxa de transmissão.

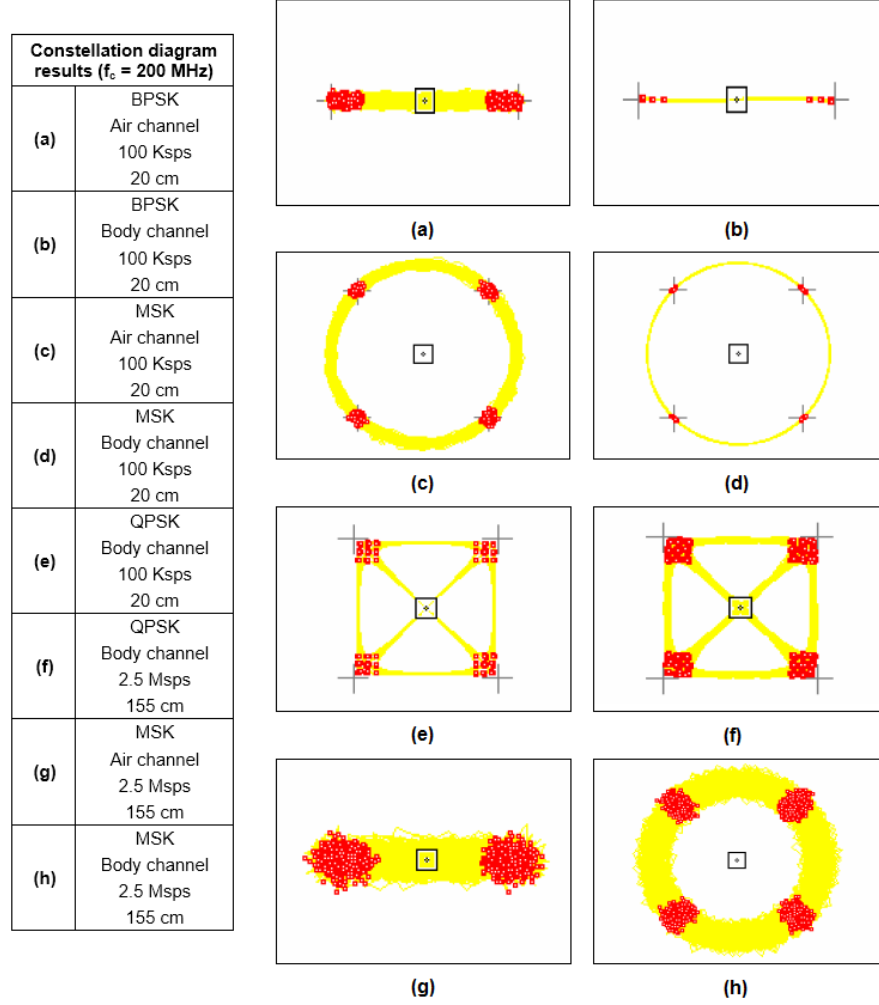


Figura 2.25. Constelações obtidas em [29].

### 2.3.6 Consumo de energia

O baixo consumo de energia é um factor crucial na implementação de um sistema BCC, pelas mais diversas razões. Segundo [30], a potência média consumida por um *transceiver* numa transmissão de dados pode ser dada por:

$$P_C = N_{tx}(P_{tx}(T_{on-tx} + T_{st}) + P_{out}T_{on-tx}) + N_{rx}P_{rx}(P_{on-rx} + T_{st}) \quad (2.1)$$

onde  $N_{tx/rx}$  representa o número de vezes que o *transceiver* é utilizado por unidade de tempo,  $P_{tx/rx}$  o consumo do *transceiver*,  $P_{out}$  a potência transmitida,  $T_{on-tx/rx}$  o tempo de transmissão/recepção e  $T_{st}$  o tempo de *start-up* do *transceiver*.

O valor de  $N_{tx/rx}$  depende da aplicação a que o sensor se destina e do mecanismo de acesso ao meio utilizado e, no caso de um sistema BCC,  $P_{out}$  é reduzido, pois pouca energia é radiada.

Outro aspecto importante prende-se com o consumo do *transceiver* relativamente ao processamento de dados, podendo este ser formulado por:

$$P_P = CV_{DD}^2 f + V_{DD} I_0 e^{V_{DD}/nV_T} \quad (2.2)$$

onde  $C$  é a capacidade total das comutações,  $V_{DD}$ , a excursão da tensão de alimentação, e  $f$ , a frequência de comutação. O segundo termo representa as perdas devido a correntes de fuga.

Como tal, o projecto de um *transceiver*, deve ter em conta a frequência de trabalho do processador e a tensão de alimentação. Pela análise da fórmula 2.2, conclui-se que a tensão de alimentação tem muito mais influência no consumo do que a frequência de trabalho do processador. Por isso, sempre que possível deve optar-se primeiro por reduzir ao máximo possível a tensão de alimentação no projecto de um *transceiver*.

Várias soluções BCC apresentadas nos últimos anos obtiveram bons resultados relativamente à taxa de transmissão mas relativamente ao consumo de energia o mesmo já não se pode afirmar. Na figura 2.26, é possível constatar este facto.

A COMPARISON OF THE IBC METHODS KNOWN SO FAR					
Method	Type	Amplitude	Frequency	Keying	Baudrate
Fukumoto	C	21 V	90 kHz	FM	0,1 kb/s
Zimmerman	C	30 V	330 kHz	OOK	2,4 kb/s
Reynolds	C	10 V	70 kHz	FSK	9,6 kb/s
Partridge	C	22 V	160 kHz	FSK	38,4 kb/s
Fujii/Ito	C	3 V	10 MHz	OOK	-
Hachisuka	C	1 V	10,7 MHz	FSK	9,6 kb/s
NTT/Docomo	C	25 V	10 MHz	OOK	10 Mb/s
Lindsey	G	3 mA	37 kHz	FM	-
Handa	G	20 $\mu$ A	70 kHz	PWM	0,9 kb/s
Oberle	G	4 mA	60 kHz	CPFSK	4,8 kb/s

Figura 2.26. Algumas soluções BCC apresentadas nos últimos anos [31].

Além destas soluções, surgiram outras que foram baseadas no uso de FPGAs ou *chipsets*, elementos estes que por sua vez também aumentam drasticamente o consumo de energia numa aplicação deste género.

No entanto, com o aparecimento de vários microcontroladores XLP (eXtreme Low Power) e de outros componentes *low-power* nos anos mais recentes, o paradigma do consumo de energia em BCC está a evoluir cada vez mais. Vêm contribuir também para este paradigma as emergentes soluções de *energy harvesting* que têm vindo a surgir nos últimos anos.

A figura 2.27 compara o consumo de energia de alguns microprocessadores tipicamente utilizados em tecnologias BCC com outros *transceivers* utilizados em tecnologias sem fios (*wireless*) comuns. No gráfico desta figura verifica-se que, no melhor dos casos (do ponto de vista

do consumo de energia), os microprocessadores *low-power* apresentados chegam a consumir até cerca de cem mil vezes menos potência que os *transceivers* das outras tecnologias *wireless*. Contudo, este cenário ainda só é alcançado com o sacrifício da taxa de transmissão.

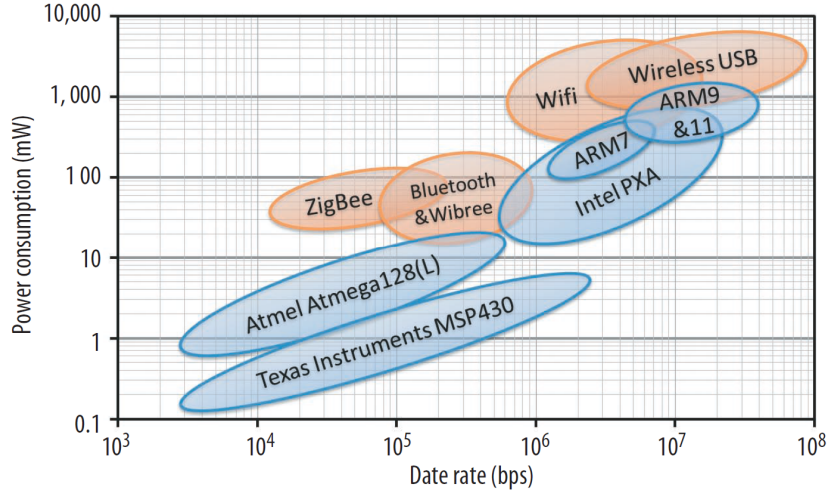


Figura 2.27. Gráfico comparativo do consumo de energia de microprocessadores utilizados em BCC e *transceivers* utilizados em tecnologias *wireless* comuns [7].

### 2.3.7 Modelo simplificado para BCC

Ao longo do tempo, diferentes abordagens foram propostas na tentativa de modelar o corpo humano como canal de transmissão. Os trabalhos realizados em [4] e [26] são exemplos disso. Contudo, estas propostas na sua maioria acabam por se revelar complexas e pouco práticas para a implementação de um sistema BCC.

Em [32] apresenta-se um modelo simplificado para BCC. Neste trabalho de investigação opta-se por uma abordagem ligeiramente diferente das demais presentes na literatura. No fundo, faz-se uma simplificação do modelo convencional dos sistemas de acoplamento capacitivo e, nesse sentido, considera-se o corpo humano como um condutor perfeito e como a referência do sistema. A figura 2.28 apresenta uma ilustração da abordagem considerada.

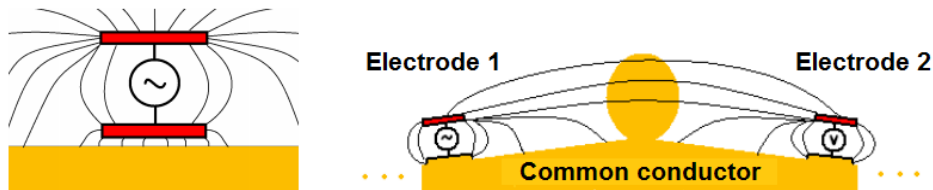


Figura 2.28. Abordagem considerada em [32].

Ao se considerar o corpo humano um plano de massa condutor, então o convencional sistema de acoplamento capacitivo da figura 2.14 pode ser simplificado e reduzido a um sistema de três condutores como se pode visualizar na figura 2.29 [32].

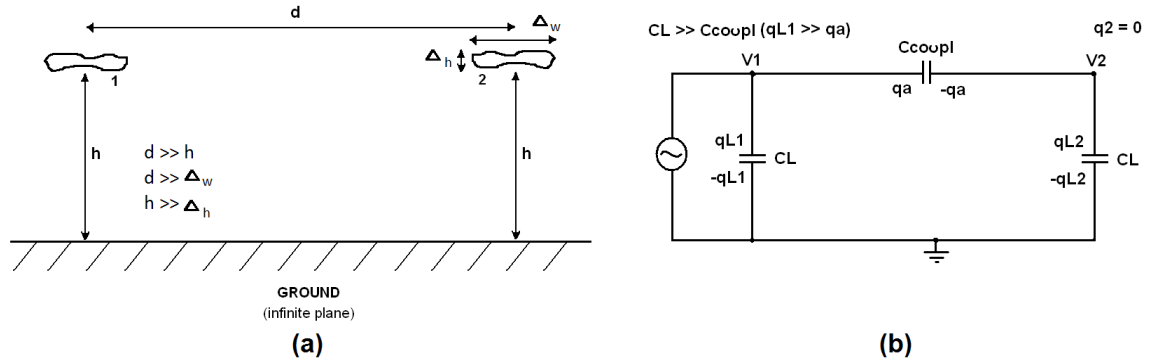


Figura 2.29. Modelo simplificado [32]. (a) Sistema de três condutores. (b) Equivalente eléctrico do sistema.

As restrições da figura 2.29 consideram que a distância entre o emissor e o receptor é muito maior do que a altura dos condutores elevados ao plano de massa e ainda, que esses condutores são muito mais finos do que a sua altura, o que permite simplificar bastante a formulação tipicamente abordada noutras implementações [32]. Deste modo, considera-se que a capacidade dos eléctrodos inferiores é muito maior que a capacidade entre os eléctrodos superiores e por isso assume-se que os eléctrodos inferiores estão curto-circuitados ao corpo para as frequências de interesse.

A partir deste modelo, derivou-se a equação que relaciona o ganho do sistema com as propriedades físicas dos eléctrodos:

$$G = 2K_e C_L \frac{h^2}{d^3} \quad (2.3)$$

onde  $K_e$  representa a constante dieléctrica do meio,  $C_L$  a capacidade do eléctrodo,  $h$  a altura dos eléctrodos em relação ao plano de referência, e  $d$ , a distância entre os eléctrodos.

Para validar o modelo proposto neste estudo, realizaram-se simulações recorrendo ao simulador de campos electromagnéticos Ansoft HFSS e construiu-se um sistema idêntico ao simulado, ao qual se efectuaram medições numa gaiola de Faraday. Na figura 2.30 apresentam-se o sistema construído e o sistema simulado.

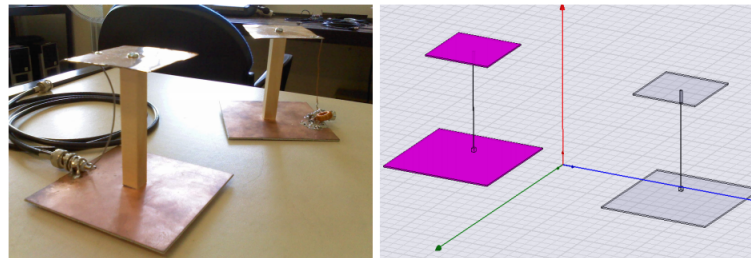


Figura 2.30. Sistema construído e sistema simulado em [32].

Na figura 2.31, apresentam-se os resultados obtidos. No caso A, o eléctrodo receptor foi ligado directamente a um cabo coaxial com capacidade de 200 pF, que por sua vez se encontrava ligado a um osciloscópio. No caso B, foi colocado um *buffer* de baixa capacidade (3 pF) entre o eléctrodo receptor e o cabo coaxial.

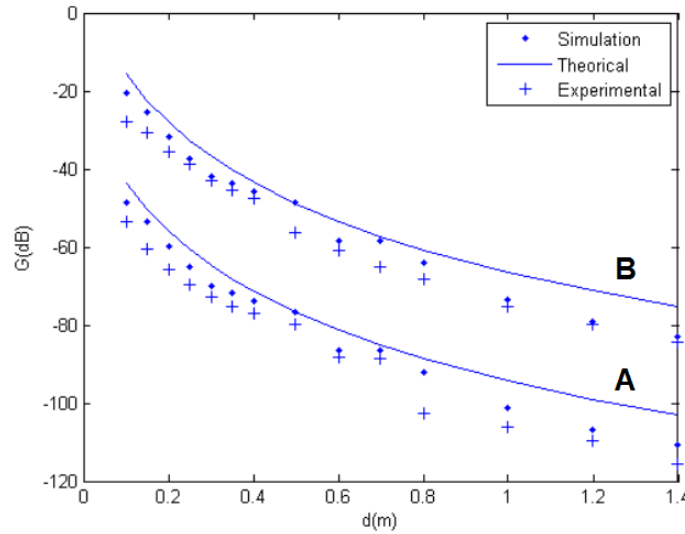


Figura 2.31. Resultados obtidos em [32].

Como se pode verificar, os resultados experimentais estão próximos dos resultados teóricos. Conclui-se ainda que a utilização de um *buffer* de entrada com baixa capacidade, introduz uma melhoria significativa no sistema implementado.

## 2.4 Sistema BCC da Universidade de Aveiro

### 2.4.1 Hugo Barreira, Universidade de Aveiro, 2009 [33]

No ano de 2009, a dissertação de mestrado de Hugo Barreira foi a primeira abordagem à construção de um sistema BCC na Universidade de Aveiro.

Hugo Barreira tentou várias implementações que foram sendo corrigidas e melhoradas ao longo do seu trabalho. No final do projecto, o mesmo conseguiu construir um protótipo que permitiu efectivamente a comunicação através do corpo humano com resultados aceitáveis. Nas figuras 2.32 e 2.33, é possível observar o diagrama de blocos do sistema e o protótipo construído, respectivamente.

Neste protótipo, utilizaram-se armações de relógios cujas bases metálicas serviam de eléctrodos para o transmissor e o receptor. Os eléctrodos foram plastificados de modo a não ficarem em contacto directo com o corpo humano. Nas experiências efectuadas utilizou-se um Timer 555 com frequência de oscilação de 10 KHz para simular a informação a transmitir. Para a modulação do sinal utilizou-se a técnica BFSK (Binary Frequency Shift Keying) com frequência centrada em 10.7 MHz.

Os resultados mostraram que foram obtidas taxas de transmissão de 10 Kbps até uma distância de 50 cm entre o emissor e o receptor. Foi observado também que, com o aumento da distância, o nível de amplitude do sinal transmitido diminuía e o ruído aumentava. A técnica de modulação utilizada e a sua respectiva frequência central de 10.7 MHz revelaram-se ambas uma boa escolha.

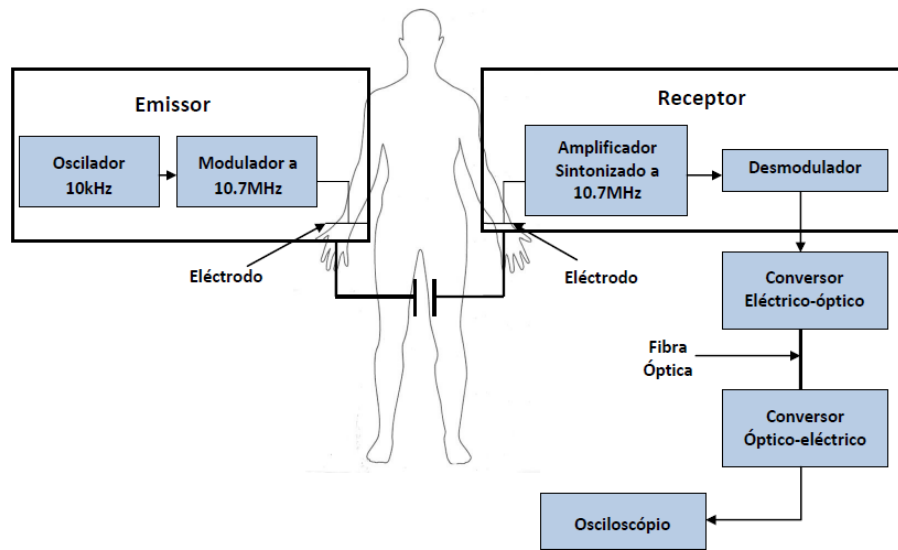


Figura 2.32. Diagrama de blocos do sistema implementado por Hugo Barreira [33].

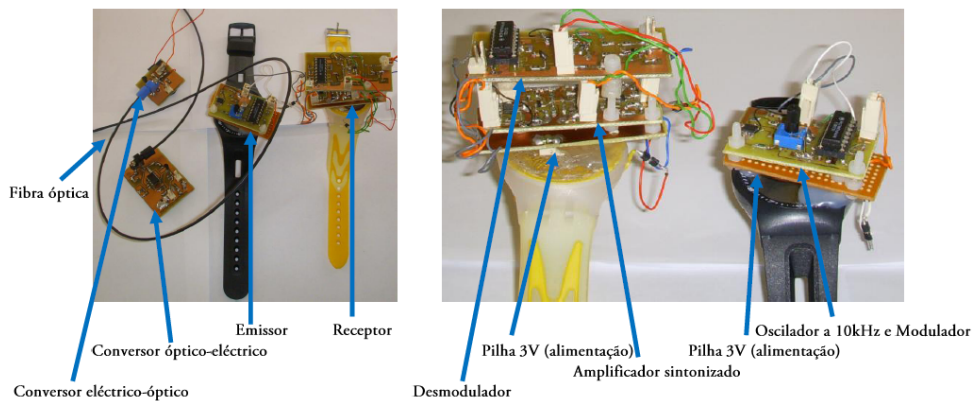


Figura 2.33. Protótipo construído por Hugo Barreira [33].

#### 2.4.2 José Reis, Universidade de Aveiro, 2010 [34]

No ano de 2010, a dissertação de mestrado de José Reis focou-se na resolução dos problemas presentes no sistema implementado por Hugo Barreira.

Um desses problemas era a fraca estabilidade do andar de entrada que introduzia distorção no sinal de entrada devido ao seu alto ganho. Inicialmente, José Reis fez uso dos programas PSpice e MATLAB para poder simular as oscilações detectadas à saída do circuito do amplificador sintonizado. Identificado o problema, procedeu ao ajuste do circuito introduzindo uma resistência na *source* do MOSFET inferior do amplificador sintonizado. Este procedimento permitiu diminuir o ganho deste andar, favorecendo a pós-amplificação em andares seguintes, e permitiu efectivamente obter maior estabilidade no sistema.



### 2.4.3 Humberto Castanheira, Universidade de Aveiro, 2012 [35]

No ano de 2012, a dissertação de mestrado de Humberto Castanheira contribuiu com importantes progressos para o desenvolvimento do sistema BCC.

Humberto Castanheira contribuiu com novas implementações, entre elas, a utilização de um *mixer* FM IF (SA636) no receptor para aumentar a sensibilidade e permitir modulação de baixa frequência e a integração de um microcontrolador *low-power* nos *transceivers* para se poder transmitir informação real. No final do projecto, construiu dois *transceivers* de dimensões reduzidas, alimentados por baterias e electricamente isolados. Os *transceivers* construídos permitiram comunicar através do corpo humano com taxas de transmissão até 100 Kbps, cobrindo praticamente toda a área do corpo e possuindo um alcance até 75 cm deste. Na figura 2.34, apresenta-se o protótipo construído por Humberto Castanheira.

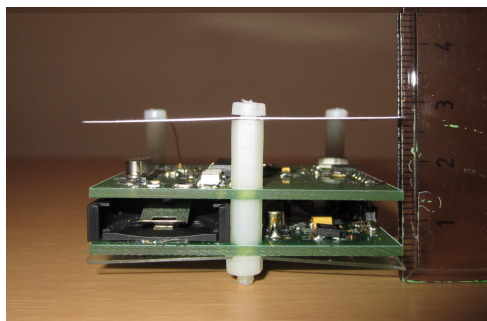


Figura 2.34. Protótipo construído por Humberto Castanheira [35].

Para demonstrar o funcionamento do sistema implementado efectuou-se uma simples experiência, na qual se configurou um *transceiver* em modo emissor e o outro em modo receptor e utilizou-se um LED que acendia sempre fossem recebidas tramas sem erros. Foi também realizada uma segunda experiência, na qual se utilizou o sistema implementado para receber um sinal ECG (simulado com recurso a um gerador de sinais) e enviar os dados recebidos para um computador que por sua vez apresentaria esses dados recorrendo ao MATLAB. Ambas as experiências permitiram comprovar a eficácia do sistema na transmissão de dados dentro dos limites impostos.

### 2.4.4 José Lima, Universidade de Aveiro, 2013 [36]

No ano de 2013, a dissertação de mestrado de José Lima teve como seu principal objectivo a implementação de um protocolo de rede que permitisse a interacção entre os vários *transceivers* numa rede. Como objectivo secundário, visava melhorar o sistema implementado por Humberto Castanheira.

José Lima fez inicialmente uma descrição detalhada do sistema, referindo também todas as novas implementações provenientes do seu trabalho.

Seguiu-se a implementação do protocolo de rede, nomeadamente as camadas PHY e MAC, tendo em conta nesta última a construção de mecanismos de acesso ao meio adequados para uma BAN. Após a implementação da camada PHY, salientou-se o facto de terem sido efectuadas modificações ao *mixer* SA636 para permitir uma melhor sintonia do circuito detector de quadratura. Na figura 2.35 é possível visualizar o *transceiver* construído por José Lima.

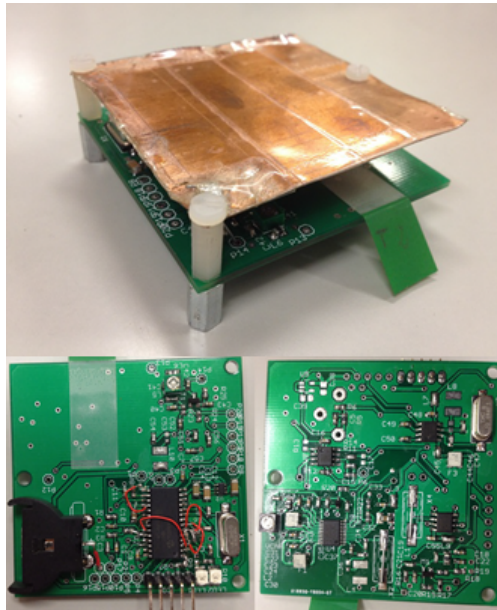


Figura 2.35. *Transceiver* construído por José Lima e respectivas PCBs. À esquerda: *top layer*. À direita: *bottom layer* [36].

Para demonstrar o sistema em funcionamento foram realizadas algumas aplicações práticas que fizessem efectivamente uso das camadas referidas. Foi desenvolvida uma aplicação de transferência de ficheiros, na qual tendo dois *transceivers* ligados cada um a um computador, foi possível realizar a transferência de ficheiros entre os dois computadores a um ritmo superior a 30 Kbps.

## 2.5 Sistemas de pagamento

### 2.5.1 Introdução

É do conhecimento geral que, desde os primórdios da história da humanidade, o homem foi sempre inventando soluções que lhe permitissem obter algo de que necessitasse ou que contribuísse para o seu benefício próprio, nomeadamente bens físicos ou serviços. Foi nesse sentido que surgiu a arte do negócio.

Antigamente, os negócios entre indivíduos eram feitos precisamente com base em trocas de bens físicos ou serviços. A mercadoria era avaliada com base na quantidade de tempo e/ou trabalho necessários para a sua produção ou com base na necessidade que o comprador tivesse pela mesma. Além disso, o negócio dependia sempre do que cada indivíduo tivesse disponível para troca.

Não é difícil de entender que esta forma de negociar tem bastantes inconvenientes e que por vezes pode mesmo tornar-se impraticável. Foi por essa razão que o homem inventou o dinheiro. Inicialmente, o dinheiro surgiu apenas em formato de moeda mas, com o decorrer do tempo, o seu conceito foi evoluindo, pelo que nos dias de hoje apresenta-se nos mais diversos formatos como exemplifica a figura 2.36.



Figura 2.36. Vários formatos de dinheiro existentes no mundo actual.

Um formato que tem emergido nos anos mais recentes é o de dinheiro digital (ou moeda electrónica). Este consiste no uso de redes computacionais em conjunto com a Internet, sistemas de armazenamento de valor digital e mecanismos extremamente sofisticados de encriptação de dados para se realizarem transacções monetárias de forma instantânea e segura entre indivíduos ou entidades em qualquer lugar do mundo. O dinheiro poderá estar depositado num dispositivo electrónico pessoal do utilizador ou numa conta digital associada a um sistema de transacções *online*, sendo por isso integralmente digital. Um exemplo de uma moeda digital (ou criptomoeda) é a famosa Bitcoin, a qual após ter sido lançada em 2009 foi tendo uma adesão cada vez maior com o passar do tempo, chegando a ter actualmente um elevado número de utilizadores na ordem das centenas de milhares ou mesmo milhões [37] [38]. A figura 2.37 mostra o logotipo habitualmente utilizado para representar esta moeda.



Figura 2.37. Bitcoin.

Um formato de dinheiro bastante comum no mundo actual e que do ponto de vista do utilizador se confunde de certo modo com o conceito de dinheiro digital, é o formato de conta bancária (ou conta à ordem). Este formato consiste no uso de dinheiro associado a um registo financeiro e mantido numa conta de depósito num banco ou noutra instituição financeira por uma pessoa física ou jurídica, de modo a proporcionar segurança e rapidez de acesso à demanda, através de uma variedade de diferentes canais [39].

Através de uma conta bancária é possível depositar/retirar dinheiro ou efectuar pagamentos um qualquer número de vezes (consoante a disponibilidade de crédito), por meio de caixas multibanco (ATM), terminais de pagamento (POS), *smartphones*, cartões de crédito, cheques ou transferências bancárias, por exemplo. Na figura 2.38 são apresentados os meios/dispositivos de pagamento mais utilizados a nível mundial.

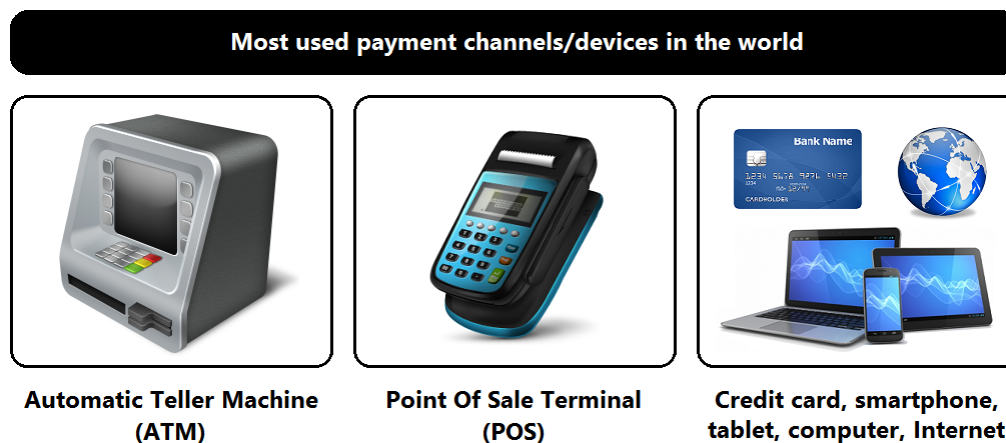


Figura 2.38. Meios/dispositivos de pagamento mais utilizados a nível mundial.

Como ilustra a figura 2.38, conjuntamente com a evolução do conceito de dinheiro, também evoluiu o acto do pagamento. Devido ao aparecimento de tecnologias cada vez mais sofisticadas, hoje em dia é possível efectuar um pagamento das mais diversas formas.

Actualmente existem muitos sistemas de pagamento e dos mais variados tipos, pelo que esta é uma área bastante vasta cuja total abrangência se encontra fora do escopo desta dissertação. Como objecto de estudo, apenas se pretende fazer a análise ao estado da arte de sistemas de pagamento do tipo *mobile*, particularmente o sistema de pagamento por NFC (ou *contactless* NFC), pelo facto de este ser uma das mais recentes inovações no paradigma dos sistemas de pagamento e porque, em termos de conceito, é o que mais se aproxima do sistema de pagamento que foi desenvolvido no âmbito desta dissertação.

### 2.5.2 Sistemas de pagamento móveis

Os sistemas de pagamento móveis (*mobile payments*) englobam todos os serviços e sistemas de pagamento cujo canal de operação é um dispositivo móvel, nomeadamente um telemóvel, um *tablet* ou um *smartphone*. Através de destes dispositivos, o utilizador pode efectuar pagamentos ou transacções monetárias em qualquer lugar do mundo, desde que existam as infraestruturas de telecomunicações necessárias. Actualmente, existem quatro principais tipos de sistemas de pagamento móveis:

- Premium SMS
- Direct Mobile Billing
- Mobile Web
- Contactless NFC

Nas secções seguintes irá introduzir-se o conceito de carteira digital (*online wallet*) e irá proceder-se à análise de cada um destes sistemas. Relativamente ao sistema *contactless* NFC irá ser feita uma análise mais detalhada, como já foi mencionado anteriormente.

### 2.5.3 Carteira digital

Uma carteira digital (ou *online wallet*) é um programa ou serviço *web* que permite ao utilizador armazenar a sua informação pessoal e verificar todas as informações relativas às suas compras e transacções *online*. Permite também que este efectue compras e pagamentos utilizando um computador, telemóvel, *tablet* ou *smartphone* com ligação à Internet. Algumas carteiras digitais já existentes são por exemplo a Google Wallet, a PayPal e a Bitcoin.

Para realizar uma compra ou efectuar um pagamento utilizando uma carteira digital, geralmente o processo é o seguinte:

No primeiro pagamento o utilizador regista-se no serviço fornecido pela entidade provedora, insere o seu número de telemóvel e a entidade envia-lhe um SMS com um PIN de autenticação. Por fim, o utilizador insere o PIN e os dados do seu cartão de crédito, validando o pagamento. Nos pagamentos seguintes, apenas é necessária a introdução do PIN para confirmar e validar os pagamentos.

Também existe a possibilidade de não ser necessária a introdução de PIN no caso de se integrar a carteira digital em plataformas móveis, sendo a autenticação efectuada por outra via, dependendo da plataforma utilizada.

Deste modo, a carteira digital é associada à conta bancária ou à conta digital do utilizador e em todos os pagamentos efectuados são debitadas da sua conta as quantias correspondentes.

### 2.5.4 Premium SMS

O modelo predominante do sistema Premium SMS consiste no requerimento do pagamento por parte do consumidor através do envio de um SMS ou um USSD com um ID único para um número especial (*short code*) fornecido pela entidade provedora dos bens físicos ou serviços. Após o envio do requerimento é debitada a quantia paga pelo utilizador na conta do seu dispositivo móvel na sua carteira digital *online*. A entidade envolvida é informada acerca do sucesso da operação, procedendo de seguida à entrega dos bens físicos ou serviços pagos pelo comprador.

Este sistema tem no entanto algumas desvantagens como a sua fraca fiabilidade, a lentidão dos processos, o alto custo e, acima de tudo, a fraca segurança. A fraca fiabilidade deve-se ao facto de as mensagens se poderem perder ou ficarem corrompidas, pois o canal de transmissão é o ar e este é um meio onde existe elevada interferência. O facto de ser lento tem a ver com a alta probabilidade de serem necessárias várias horas para o comerciante receber o recibo do pagamento e poder assim concluir o processo da compra. O alto custo é devido às despesas associadas à criação *short codes* e suporte para o cliente e também às despesas associadas aos próprios SMS/USSD. A fraca segurança deve-se ao facto de, mais uma vez, os SMS/USSD serem enviados pelo ar via RF, ou seja, a probabilidade de as mensagens serem interceptadas é alta. Além disso, a encriptação deste tipo de mensagens é também bastante fraca.

### 2.5.5 Direct Mobile Billing

O sistema de pagamento Direct Mobile Billing (facturamento móvel directo) consiste na utilização de uma opção de facturamento móvel por parte do cliente durante o acto do pagamento num qualquer *site* de comércio electrónico (*e-commerce*), onde o canal utilizado é um dispositivo móvel com acesso à Internet. Não requer a utilização de cartões de crédito nem o pré-registo em *sites* de soluções de pagamento como o PayPal por exemplo, pelo que apenas é necessária uma TFA (Two-Factor Authentication) envolvendo o uso de um PIN e uma OTP (One-Time Password). Depois de o cliente se autenticar com sucesso, é debitada da sua conta móvel a quantia paga pela compra efectuada.

Este sistema tem diversas vantagens como a forte segurança, conveniência, simplicidade e rapidez nos processos. É seguro devido ao facto de se munir com uma TFA, o que ajuda bastante à prevenção de fraudes. É conveniente e simples por não ser necessário qualquer pré-registo em *sites* de pagamentos nem o uso de *software* específico, sendo apenas necessária a selecção da respectiva opção no processo de *checkout*. É rápido pelo facto de os processos se completarem numa questão de segundos.

### 2.5.6 Mobile Web

Num sistema de pagamento Mobile Web (rede móvel), o cliente utiliza páginas *web* ou aplicações móveis específicas para efectuar um pagamento através do seu *tablet* ou *smartphone*. Como tal, o dispositivo utilizado necessita de ter uma ligação à Internet. Neste sistema é necessário o uso de um cartão de crédito ou o pré-registo num *site* de soluções de pagamento. Tem ainda a particularidade de utilizar WAP (Wireless Application Protocol), o qual é um *standard* específico para o acesso a informação em redes móveis *wireless*, permitindo a oferta de uma enorme variedade de serviços interactivos. No entanto, até aos dias de hoje, o WAP tem entrado em desuso na Europa e nos Estados Unidos em prol do HTML (HyperText Markup Language) [40].

Dentro deste sistema existem três métodos de pagamento possíveis, sendo eles, Direct Operator Billing, cartão de crédito e carteira digital. No primeiro método, é utilizado um sistema semelhante ao de Direct Mobile Billing, no qual os pagamentos são debitados na conta móvel do utilizador. No segundo método, é solicitada a inserção de alguns detalhes do cartão de crédito do utilizador de forma a este poder ser identificado e para ser debitada a quantia paga da conta associada ao seu cartão. No terceiro método, o utilizador associa a sua carteira digital ao processo de pagamento, pelo que apenas necessita de introduzir um PIN para se autenticar e efectuar o pagamento.

As principais vantagens do sistema Mobile Web são a sua segurança, simplicidade e rapidez nos processos. É seguro pelo facto de serem usados diversos mecanismos de autenticação e encriptação consoante o método de pagamento utilizado. É simples e rápido porque através do uso de páginas *web* ou aplicações móveis a realização de uma compra e do seu respectivo pagamento tornam-se bastante simples, processo este que normalmente dura poucos segundos.

### 2.5.7 Contactless NFC

O sistema de pagamento por NFC (ou *contactless* NFC) é um sistema que foi concebido para ser utilizado principalmente em lojas físicas, transportes ou em qualquer outra entidade física que comercialize bens físicos ou serviços. O utilizador deve possuir um dispositivo móvel (*smartphone* por exemplo) ou um cartão de crédito ambos equipados com um *smartcard* que permite o uso da tecnologia NFC. Para efectuar um pagamento basta simplesmente ao utilizador aproximar o seu dispositivo móvel ou cartão de crédito a um terminal de pagamentos, o qual consiste num leitor NFC baseado em tecnologia RFID, sendo apenas necessária a introdução de um PIN para autenticação.

Existem também a possibilidade de o utilizador poder pré-definir a não introdução de PIN para pagamentos de baixas quantias. Neste caso, o mesmo terá um saldo diário limitado para efectuar pagamentos, evitando assim burlas graves. Contudo, será diminuída a segurança ao se optar por uma via deste género.

No caso de ser utilizado um *smartphone* como meio de pagamento, o dispositivo terá associada a carteira digital ou a conta bancária do utilizador, pelo que as quantias pagas serão automaticamente debitadas numa destas.

Este sistema de pagamento tem diversas vantagens como a sua forte segurança, conveniência, simplicidade e rapidez nos processos. É seguro porque são utilizados canais seguros com forte encriptação para enviar as todas as informações, além das próprias características da tecnologia NFC que só por si garantem maior segurança pelo facto de a comunicação ser de curto alcance. É simples, rápido e cómodo por ser apenas necessário tocar ou aproximar o dispositivo ao terminal NFC para efectuar um pagamento e o processo de completação dura geralmente poucos segundos. De seguida, irá ser feita uma descrição mais detalhada da tecnologia e dos processos envolvidos neste sistema de pagamento.

O NFC (Near-Field Communication) é um conjunto de *standards* para *smartphones* e dispositivos similares cuja função é estabelecer comunicação RF de curto alcance entre estes. Para estabelecer a comunicação apenas é necessária a aproximação ou o toque entre os dispositivos. A figura 2.39 mostra o logotipo habitualmente utilizado para representar a tecnologia NFC.



Figura 2.39. NFC.

O conceito de NFC teve origem por volta de 2002, através de um acordo entre a Sony e a Philips com o intuito de estabelecer especificações relativamente a uma nova tecnologia que viria a ser uma variante do HF RFID. Em 2004 e em conjunto com a Nokia, criaram o NFC Forum para facilitar e dar continuidade ao processo de definição dos padrões do NFC [41]. Este fórum conta actualmente com um elevado número de membros, nos quais estão incluídas várias empresas multinacionais.

Em 2006, surge o primeiro telemóvel equipado com NFC (Nokia 6131) e mais tarde, no ano de 2010, surge o primeiro *smartphone* Android também equipado com NFC (Samsung



Google Nexus S). A partir daí, várias entidades de todo o mundo dos mais diversos sectores começaram a disponibilizar bens físicos e serviços que fazem uso desta tecnologia. Uma das novidades foi a introdução de pagamentos via *contactless* NFC.

Existem diversas razões pelo facto de o NFC ser utilizado para sistemas de pagamento. Primeiro que tudo, esta é uma tecnologia de alcance extremamente reduzido (cerca de 10 cm no máximo), o que só por si garante maior segurança num sistema deste tipo e reduz a interferência entre dispositivos que se encontrem nas proximidades. Depois, permite efectuar comunicações sem fios (*wireless*) entre dispositivos com taxas de transmissão mais que suficientes para sistemas de pagamento (até 424 Kbps em complacência com a norma ISO/IEC 18092). Além disso, o facto de poder ser integrada em *smartphones* ou cartões de crédito é de extrema conveniência para o utilizador geral, pois torna-se muito mais prática e cómoda a efectuação de um pagamento simplesmente por aproximação de um *smartphone* ou de um cartão de crédito a um terminal, como ilustra a figura 2.40.



Figura 2.40. Formas de pagamento via *contactless* NFC.

Alguns dos sistemas de pagamento *contactless* NFC já existentes são, por exemplo, o Visa payWave ou o MasterCard PayPass.

O NFC tem-se tornado cada vez mais popular até aos dias de hoje. No ano de 2013, o número de pagamentos utilizando cartões de crédito Visa *contactless* NFC quase quadruplicou no Reino Unido relativamente ao ano de 2012, chegando a atingir a marca dos 94.3 milhões. Nesse período, os cinco maiores mercados da Europa para os pagamentos efectuados por esta via foram a Polónia (158.7 milhões), o Reino Unido (94.3 milhões), a República Checa (30.6 milhões), a Eslováquia (23.1 milhões) e a Espanha (17.5 milhões). Por sua vez, a quantidade de cartões Visa *contactless* NFC em circulação por toda a Europa cresceu 158%, alcançando os 80.6 milhões [42]. Além disto, a Visa Europe prevê que por volta do ano 2020, cerca de 50% das transações na sua rede serão efectuadas via *smartphone* ou outros dispositivos móveis [43].

A importância do NFC no âmbito dos sistemas de pagamento tem sido cada vez maior, pelo que a Samsung e a Visa anunciaram em 2013 uma enorme parceria com o objectivo de desenvolverem novos sistemas de pagamento *mobile* centrados nesta tecnologia [44]. De igual modo, em 2014 a MasterCard e a Weve (*joint venture* entre a EE, a Telefónica UK e a Vodafone UK) também anunciaram uma parceria nesse sentido e com o objectivo de criarem uma plataforma universal na Europa [45].

O NFC é uma norma que é habitualmente confundida com o RFID. Apesar de ser baseado neste, existem algumas diferenças. Como já se referiu anteriormente, o NFC é uma variante do HF RFID. A tabela 2.5 apresenta as principais diferenças entre estas duas tecnologias.



	NFC	HF RFID
Frequência	13.56 MHz	13.56 MHz
Alcance	< 10 cm	< 1 m
Comunicação	Bidireccional	Unidireccional
Leitura simultânea de várias tags	Não	Sim
Normas	ISO/IEC 14443, 18092	ISO/IEC 14443, 15693, 18000

Tabela 2.5. Principais diferenças entre o NFC e o HF RFID.

Por sua vez, nas tabelas 2.6 e 2.7, apresentam-se algumas das principais características da tecnologia NFC.

#### Características da tecnologia NFC

Modos de comunicação	Read/Write, Card Emulation, P2P
Modos de operação	Activo/Passivo
Frequência	13.56 MHz
Largura de banda	7 KHz - 1.8 MHz
Alcance	< 10 cm
Taxas de transmissão	106 Kbps, 212 Kbps, 424 Kbps
Tempo de setup	< 0.1 s
Corrente de pico	< 15 mA (Read)
Corrente de standby	< 1 $\mu$ A

Tabela 2.6. Características da tecnologia NFC.

#### Codificações e modulações utilizadas em NFC

Taxa de transm.	Modo activo	Modo passivo
106 Kbps	Miller modificado, 100% ASK	Manchester, 10% ASK
212 Kbps	Manchester, 10% ASK	Manchester, 10% ASK
424 Kbps	Manchester, 10% ASK	Manchester, 10% ASK

Tabela 2.7. Codificações e modulações utilizadas em NFC.

Como se pode observar pelas tabelas 2.6 e 2.7, existem dois modos de operação em NFC, sendo eles o modo activo e o modo passivo.

No modo activo, ambos os dispositivos *initiator* e *target* têm as suas próprias fontes de alimentação e comunicam gerando alternadamente os seus próprios campos electromagnéticos. Quando um dispositivo aguarda pela recepção de dados, o mesmo não gera qualquer campo electromagnético.

No modo passivo, o dispositivo *initiator* gera um campo electromagnético e o dispositivo *target* responde modulando esse campo. Na maioria dos casos, apenas o *initiator* necessita de ter a sua própria fonte de alimentação, pelo que o *target* aproveita a energia proveniente do campo electromagnético gerado pelo *initiator* para realizar as suas operações. Por esta razão, o dispositivo *target* é várias vezes designado por *transponder*. Alguns dos dispositivos que comunicam neste modo são, por exemplo, as NFC *tags* (bastante utilizadas em identificação de objectos) e os *contactless smartcards* utilizados em sistemas de pagamento.

As figuras 2.41 e 2.42 ilustram o funcionamento dos modos de operação activo e passivo em NFC.

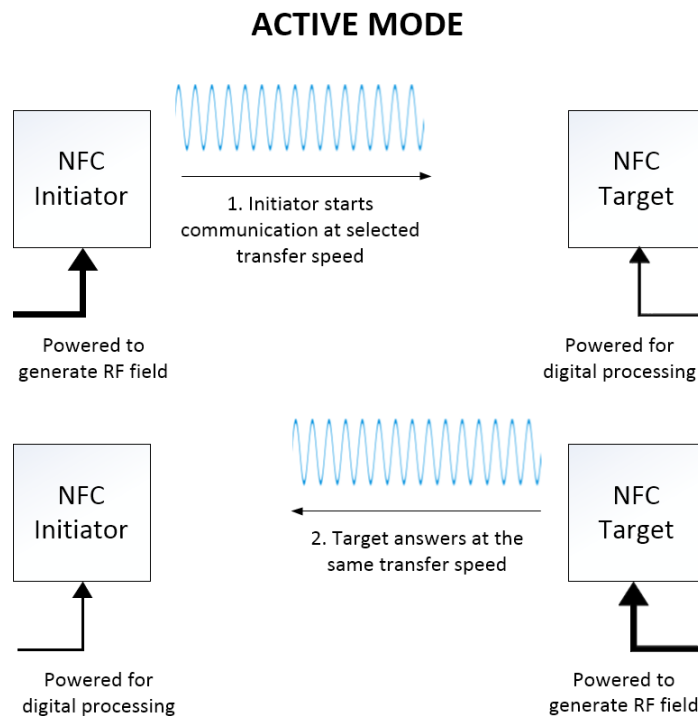


Figura 2.41. Modo de operação activo em NFC. Baseado em [46].

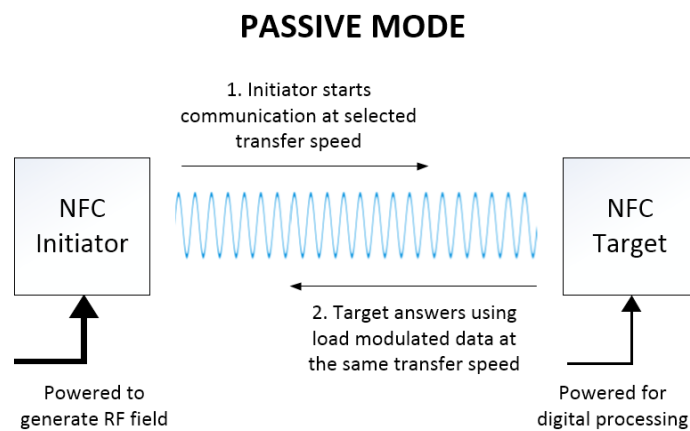


Figura 2.42. Modo de operação passivo em NFC. Baseado em [46].

Relativamente aos modos de comunicação, pela tabela 2.6 é possível constatar que existem três, sendo eles Read/Write, Card Emulation e P2P (*peer-to-peer*).

O modo Read/Write permite a transmissão de mensagens do tipo definido pelo NFC Forum, podendo ser utilizado para a leitura de NFC *tags*, por exemplo. O modo Card Emulation permite que um dispositivo (*smartphone* por exemplo) se comporte como um *contactless smartcard*. O modo P2P (*peer-to-peer*) é definido para comunicação dispositivo-dispositivo (*link-level*), permitindo a troca de mensagens e dados entre dispositivos. Dos três, o modo Card Emulation é o único que é seguro e suportado pela Contactless Communication API (cinco pacotes Java que definem um conjunto de APIs para a proximidade e comunicação *contactless*). A figura 2.43 apresenta as diferenças entre os três modos de comunicação e as relações destes com as normas NFC.

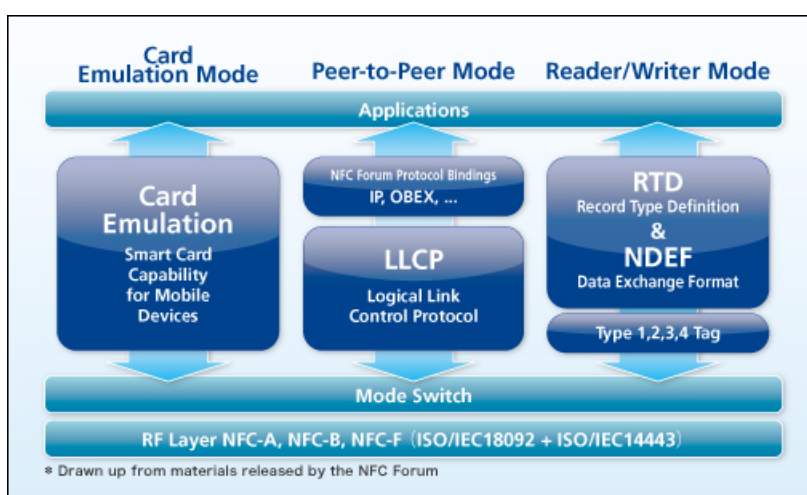


Figura 2.43. Modos de comunicação em NFC [47].

A norma ISO 14443 é particularmente direccionada para os *contactless smartcards*. A norma ISO 18092 define os modos de comunicação para a interface e para o protocolo NFC. A funcionalidade Mode Switch é responsável pela gestão do processamento dos três modos de comunicação.

Dos três modos de comunicação, o que é utilizado nos sistemas de pagamento é o modo Card Emulation. Este modo é automaticamente activado no chip NFC do *smartphone* assim que se aproxima este último ao terminal de pagamentos. O seu funcionamento consiste na comunicação via *hardware* RFID do elemento seguro (*smartcard*) do dispositivo (*smartphone* neste caso) com um leitor NFC externo. Durante o processo, as aplicações são notificadas acerca das transacções mas não têm qualquer participação neste. A figura 2.44 ilustra este cenário.

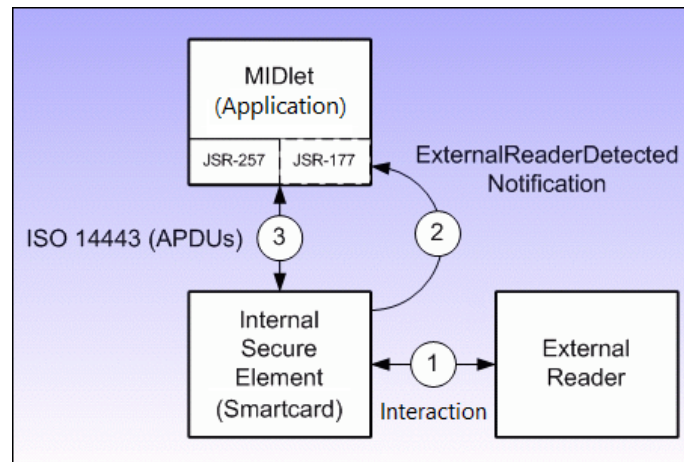


Figura 2.44. Funcionamento do modo Card Emulation. Baseado em [48].

A comunicação entre o elemento seguro e o leitor externo é transparente à aplicação e ocorre usando comandos APDU (definidos na norma ISO 7816-4) similarmente à forma como os Java Cards comunicam com leitores externos. Para a comunicação entre a aplicação (MIDlet) e o elemento seguro (*smartcard*) é necessário o conhecimento interno do aplicativo (*applet*) do elemento seguro [48]. A figura 2.45 ajuda a compreender melhor este processo apresentando um caso genérico. Note-se que num dispositivo NFC, a MIDlet poderá ser a própria MIDlet, um leitor interno ou um leitor externo (via modo Card Emulation). Por sua vez, o elemento seguro poderá ser interno ou externo e é acessível via JSR-177 (SATSA - Security And Trust Services API for J2ME), JSR-257 (Contactless Communication API) ou via *hardware* RFID.

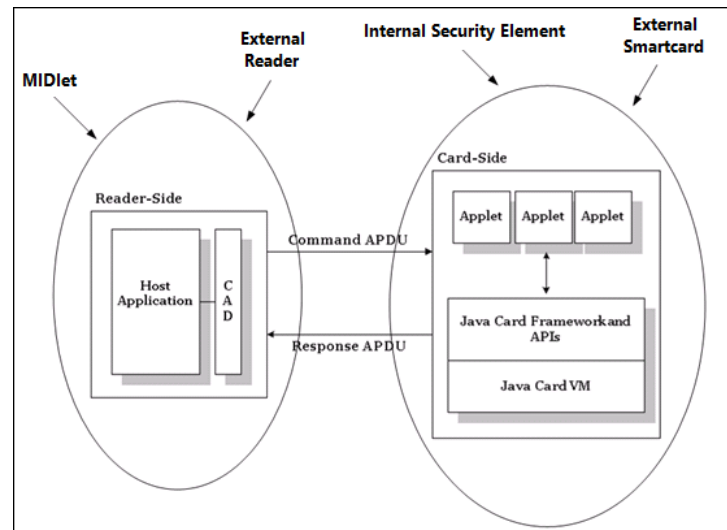


Figura 2.45. Exemplo genérico de uma comunicação entre uma aplicação (MIDlet) e um *smart-card* (Security Element) [48].

## Capítulo 3

# Sistema de comunicação

### 3.1 Sumário

No presente capítulo descreve-se o sistema BCC implementado no âmbito deste projecto. É feita uma breve introdução à arquitectura do *transceiver* construído, procedendo-se de seguida à descrição detalhada dos vários blocos que constituem o sistema. Todas as implementações que foram mantidas de anos anteriores são referenciadas e analisadas. De igual modo, é também feita a análise a todas as novas implementações e melhorias efectuadas ao sistema.

Depois de integrado no sistema de pagamento, o sistema BCC terá como finalidade permitir a comunicação através do corpo humano entre o *smartphone* e o terminal de pagamentos.

### 3.2 Arquitectura do sistema de comunicação

Como já foi referido anteriormente, o conceito da tecnologia BCC difere bastante do conceito das tecnologias RF convencionais. De igual modo, fez-se menção às suas diferentes regiões de radiação electromagnética *near-field* e *far-field*, respectivamente.

Nos sistemas RF, o elemento que transmite a informação é a antena. Esta é habitualmente vista como uma impedância, a qual por sua vez tem relação directa com a eficiência de radiação. Durante o processo de radiação, os sinais eléctricos são convertidos em ondas electromagnéticas cujo meio de propagação passa a ser o ar, tipicamente. Dependendo das dimensões da antena, das suas frequências de trabalho e da potência de transmissão, o alcance da transmissão pode situar-se tanto num raio de apenas alguns metros como num raio de vários quilómetros.

Por sua vez, num sistema BCC o elemento que transmite a informação é o eléctrodo. De certo modo, este pode ser visto como a antena do sistema. No entanto, o funcionamento do mesmo é completamente diferente do das antenas dos sistemas RF, pelo que o eléctrodo é habitualmente visto como um condensador, ou seja, um circuito aberto. Devido às suas dimensões reduzidas, às suas baixas frequências de trabalho e também à sua reduzida potência de transmissão, o processo de radiação é extremamente ineficiente, pelo que o alcance da transmissão acaba por ser bastante curto quando o ar é o meio de propagação. Ao contrário dos sistemas RF, esta característica é precisamente o que se pretende para um sistema BCC. Não é demais referir que para um sistema de pagamento BCC acaba mesmo por ser crucial, pelas razões já mencionadas anteriormente.

Apesar das claras diferenças entre as tecnologias RF e BCC, a arquitectura dos sistemas de comunicação de ambas é semelhante. Na figura 3.1, é apresentada a arquitectura básica de um sistema de comunicação.

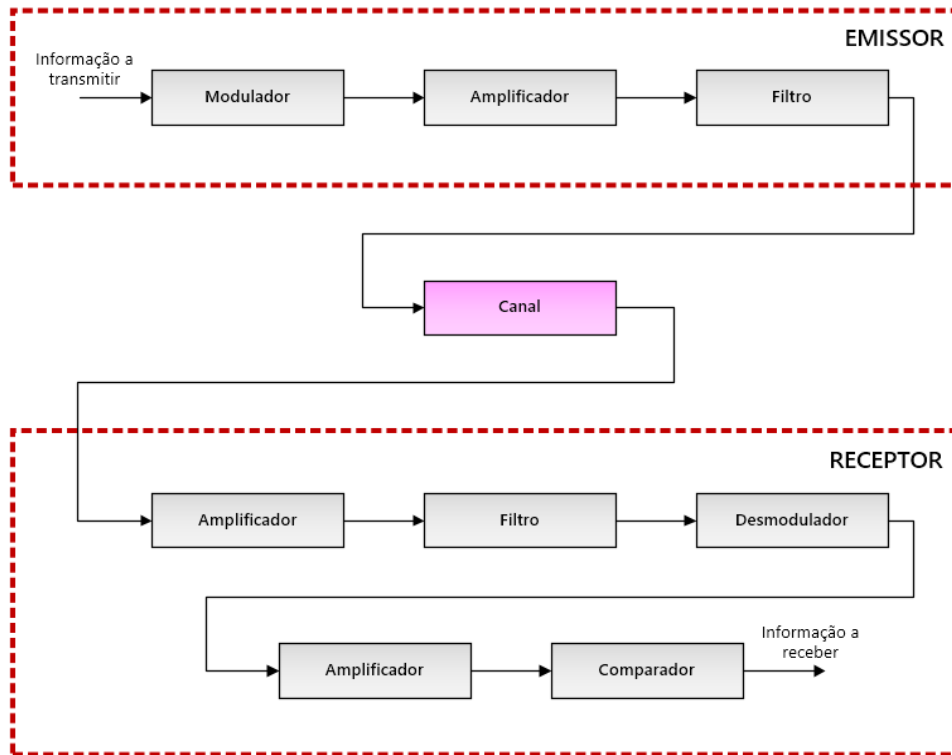


Figura 3.1. Arquitectura básica de um sistema de comunicação.

No caso do sistema BCC implementado, o emissor e o receptor são os *transceivers* desenvolvidos e o canal de transmissão é logicamente o corpo humano, sendo a informação transmitida por acoplamento capacitivo a este último. A figura 3.2 apresenta a arquitectura do sistema de comunicação implementado.

Esta arquitectura foi fruto do trabalho realizado em [36], pelo que no âmbito deste projecto de dissertação foram efectuadas apenas algumas modificações e melhorias ao nível do *hardware* e do *firmware*. Não houve necessidade de se proceder à alteração da sua estrutura, pois a mesma apresentou bons resultados no ano anterior.

Relativamente ao *hardware*, uma das modificações foi a substituição do microcontrolador da versão anterior do *transceiver* (Microchip PIC24HJ64GA102) por um microcontrolador XLP (eXtreme Low Power) mais recente e com mais funcionalidades. O microcontrolador escolhido foi o Microchip PIC24FJ128GA306. Uma das características mais importantes deste microcontrolador é o facto de consumir menos potência que o anterior. De modo a possibilitar a integração do novo microcontrolador no *transceiver* actual, foram remapeados todos os pinos do microcontrolador anterior para o novo microcontrolador e foi efectuada a sua reprogramação com o novo *firmware*.

As restantes modificações efectuadas ao nível do *hardware* irão ser discutidas nas secções seguintes, particularmente nas secções 3.7 e 3.8.

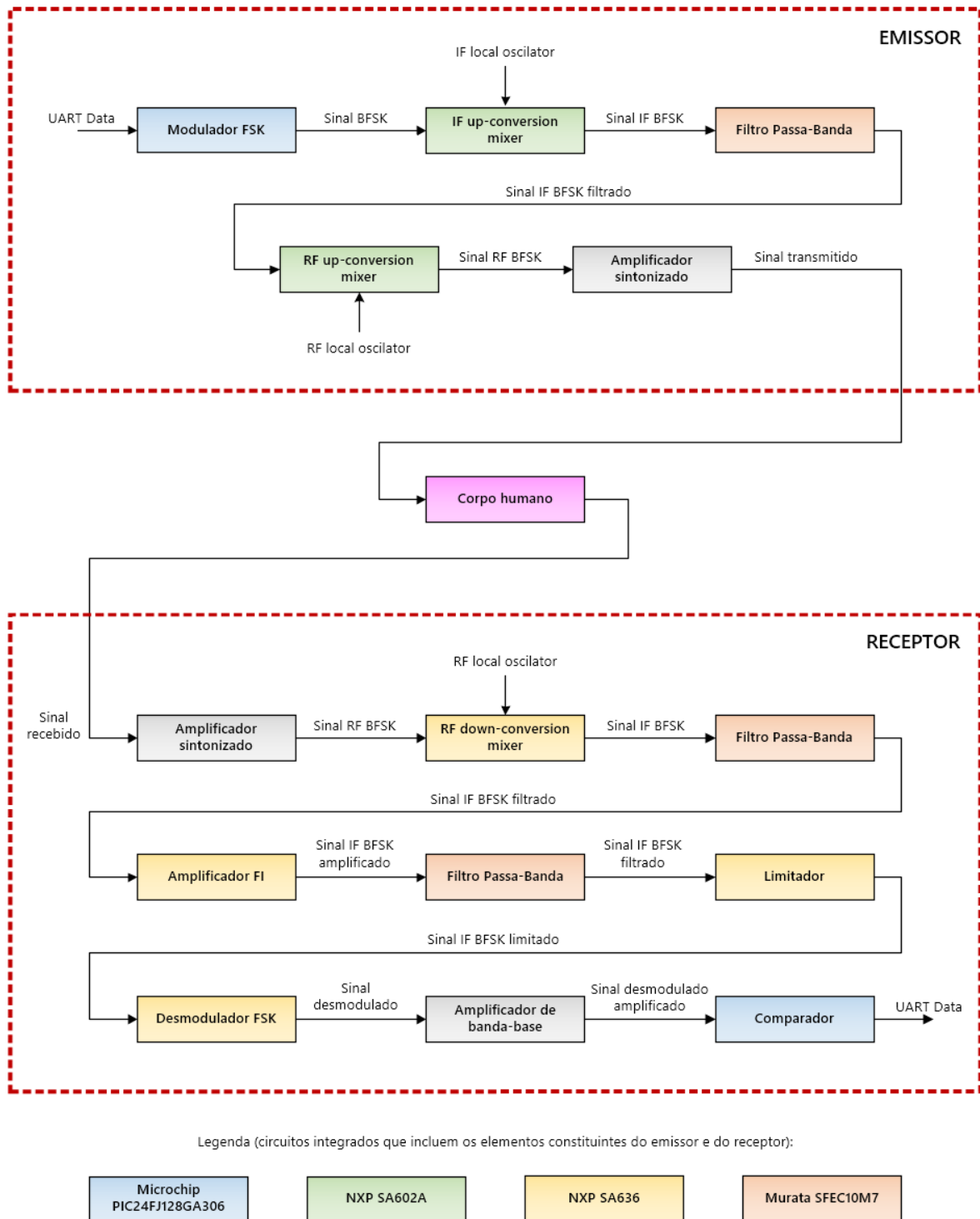


Figura 3.2. Arquitectura do sistema de comunicação implementado.

Como se pode observar pela figura 3.2, a informação a transmitir é gerada pelo microcontrolador a partir de um dos seus módulos UART. Note-se que o sinal original pode ser tanto analógico como digital e pode ter origem externa ou interna. De seguida, e ainda no micro-

controlador, é gerado o sinal BFSK de banda-base (quadrado) com frequências correspondentes aos símbolos 0 e 1 de 666 KHz e 800 KHz, respectivamente. Este sinal é depois transladado para a frequência central de 10.7 MHz através do *mixer* NXP SA602A (cuja frequência de oscilação local é de 10 MHz), pelo que as frequências correspondentes aos símbolos 0 e 1 passam a ser 10.666 MHz e 10.800 MHz, respectivamente. Devido à sua elevada largura de banda (um sinal quadrado tem largura de banda idealmente infinita devido aos seus harmónicos), o sinal é filtrado pelo filtro passa-banda cerâmico de alta selectividade (Murata SFEC10M7) com frequência central de 10.7 MHz, de modo a excluir os harmónicos indesejados e apenas a reter as frequências fundamentais das portadoras correspondentes aos símbolos 0 e 1. Depois, dependendo da aplicação, o sinal pode ser transladado para outra frequência de interesse através do *mixer* NXP SA602A novamente. Por fim, é amplificado e transmitido pelo corpo humano por acoplamento capacitivo.

Ao chegar ao receptor, o sinal é amplificado (pois sofre atenuação no corpo humano) e, através do *mixer* do receptor FM NXP SA636, é convertido novamente para um sinal BFSK de frequência intermédia (10.7 MHz) de modo a ser tratado de forma mais conveniente. De seguida, é filtrado por um filtro igual ao do emissor para se obter apenas a banda de interesse e depois é novamente amplificado. Como o amplificador FI possui alto ganho, o sinal tem de ser filtrado novamente e os seus níveis máximos têm de ser limitados por um limitador. A limitação do sinal não tem qualquer influência na recepção pois o factor de interesse encontra-se nas transições por zero e não na amplitude do sinal. Depois de limitado, o sinal é desmodulado por um detector de quadratura (evitando-se assim a realização de um processo inverso ao do emissor) e amplificado logo de seguida devido à sua baixa excursão à saída do detector de quadratura. Tanto o amplificador FI como o limitador e o detector de quadratura estão presentes no NXP SA636. Por fim, o sinal digital é obtido utilizando um circuito de decisão, o qual consiste num comparador integrado no microcontrolador.

É importante referir que existiram duas razões principais para a escolha da frequência portadora de 10.7 MHz. Uma foi o facto de o estudo realizado em [28] concluir que o corpo humano apresenta a menor atenuação para esta frequência e a outra foi o facto de esta ser tipicamente a frequência intermédia dos receptores FM super-heteródinos comerciais.

Os *transceivers* do sistema de comunicação implementado são baseados precisamente na arquitectura dos receptores FM super-heteródinos. No receptor, o andar RF *down-conversion mixer* tem a função de converter o sinal RF BFSK para um sinal de frequência intermédia IF BFSK, permitindo deste modo um tratamento do sinal mais conveniente. Por sua vez, no emissor o andar FI *up-conversion mixer* tem como objectivo converter o sinal BFSK de banda-base num sinal de frequência intermédia IF BFSK, facilitando o seu tratamento ao permitir um confinamento mais eficiente do espectro nas posteriores filtrações. De igual modo, o andar RF *up-conversion mixer* do emissor permite converter o sinal IF BFSK para um sinal RF BFSK com outra frequência intermédia. Como pequena nota, refere-se que foi em [36] que foram introduzidos pela primeira vez os blocos RF *up-conversion mixer* do emissor e RF *down-conversion mixer* do receptor, os quais permitem a utilização de outras frequências de trabalho para além da frequência de 10.7 MHz. O estudo do emissor e do receptor será realizado com mais detalhe nas secções 3.5 e 3.6 deste capítulo.



### 3.3 Microcontrolador

A função do microcontrolador neste sistema de comunicação é, como o próprio nome indica, controlar todo o *hardware* presente no sistema. Neste caso, o microcontrolador escolhido deverá possuir os seguintes módulos:

- 1 módulo ADC com pelo menos 4 canais para ler os níveis da RSSI, da bateria, do sinal médio de recepção e de uma entrada analógica externa.
- 2 módulos Output Compare para gerar o sinal FSK e o sinal PWM necessário para o *tuning* do *varicap* do desmodulador FSK.
- 2 módulos UART para a transmissão/recepção do *transceiver* e também para a comunicação com o PC.
- 1 módulo Change Input Notification para a decisão das frequências a gerar pelo Output Compare de acordo com o bit transmitido pela UART.
- 1 módulo Comparator com nível de referência programável para o circuito de decisão do sinal desmodulado.
- 1 módulo Comparator para detectar transições no nível da RSSI acima de um limiar pré-programado.
- 1 módulo Timer para o processamento dos pacotes recebidos.

Como já foi mencionado anteriormente, o microcontrolador escolhido para o sistema de comunicação foi o Microchip PIC24FJ128GA306, pois possui todos estes módulos. Além disso, funciona com tensões de operação entre 2.0V e 3.6V e possui dois reguladores de tensão que permitem operar em regime XLP com tensões de 1.2V e 1.8V, proporcionando baixos consumos. No *transceiver*, o microcontrolador é alimentado a 3V, algo que será explicado em mais detalhe na secção 3.7 (Alimentação). Para o correcto funcionamento do *transceiver* foi desenvolvido e programado o *firmware* necessário no microcontrolador. O *firmware* foi desenvolvido em linguagem de programação C utilizando o programa MPLAB X.

### 3.4 Sincronismo UART

Em qualquer sistema de comunicação é necessária a existência de um código de linha ou protocolo que permita o sincronismo entre o emissor e o receptor de modo a evitar que ocorram erros na comunicação. De igual modo, o sincronismo é um factor essencial para a correcta comunicação entre os *transceivers* do sistema implementado neste projecto.

Uma solução possível para este sistema seria a implementação de uma comunicação síncrona, na qual existiria informação acerca do relógio internamente em ambos o emissor e o receptor. Outra solução seria a implementação de uma comunicação assíncrona, na qual a informação acerca do relógio seria transmitida em conjunto com os dados. No caso de uma comunicação síncrona, o código de linha a utilizar pode ser o mais simples (NRZ), pois basta ter-se conhecimento do ritmo de transmissão de dados. No caso de uma comunicação assíncrona, apesar de não ser necessário o conhecimento do ritmo de transmissão de dados, já é necessária a utilização de códigos de linha mais sofisticados, como por exemplo o código de Manchester,

o que por sua vez aumenta consideravelmente a complexidade de um sistema de comunicação. Nos *transceivers* implementados existe apenas um canal de comunicação (frequência portadora), pelo que se for pretendida a transmissão da informação relativa ao relógio, esta dependerá do código de linha utilizado [36].

Por outro lado, a utilização de tramas UART permite comunicações assíncronas sem a necessidade de se transmitir a informação relativa ao sinal de relógio. Em vez disso, são enviados bits de sincronismo (start/stop bits) nas tramas de dados para recuperar a informação do sinal de relógio. Tem ainda a vantagem de poupar no consumo de energia, pois quando não existem dados para enviar, nenhum bit de sincronismo é enviado, ao contrário das comunicações síncronas. Portanto, no sentido de simplificar a comunicação na banda base entre os *transceivers*, optou-se pela utilização de tramas UART. Deste modo, configurou-se a trama UART da seguinte forma: 1 start bit (ST), 8 data bits (D1-D7), no parity, 1 stop bit (SP).

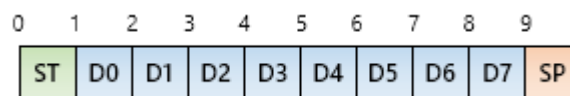


Figura 3.3. Trama UART configurada.

Contudo, existe uma desvantagem na utilização das tramas UART, como se pode reparar. Esta deve-se ao *overhead* introduzido pelos bits de sincronismo, que conduz a que 20% da largura de banda seja desperdiçada apenas para sincronizar as tramas.

## 3.5 Emissor

A figura 3.4 apresenta o diagrama de blocos do emissor.

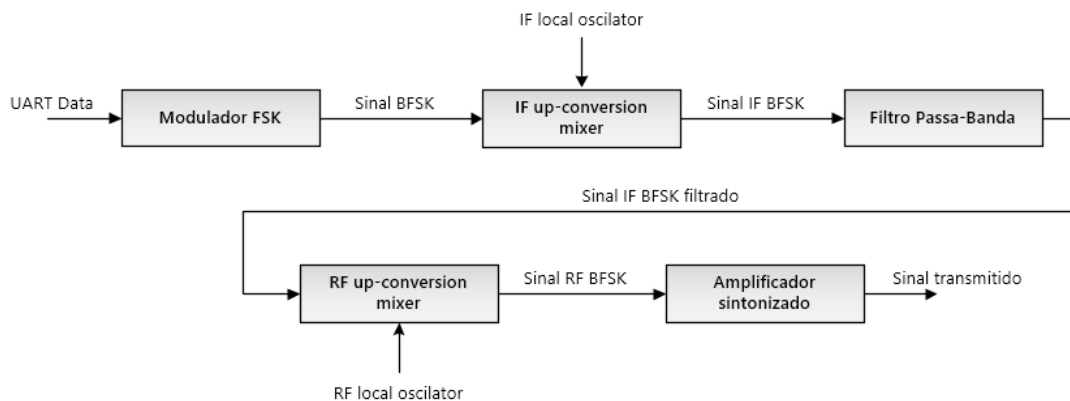


Figura 3.4. Diagrama de blocos do emissor.

Nas secções seguintes serão analisados em detalhe os vários blocos do emissor.

### 3.5.1 Modulador FSK

De modo a poder ser enviado pelo canal de transmissão (corpo humano), o sinal a transmitir primeiro necessita de ser modulado. A modulação utilizada no sistema de comunicação implementado foi a FSK (Frequency-Shift Keying), mais especificamente, a BFSK (Binary

Frequency-Shift Keying) pois apenas são transmitidos os símbolos binários 0 e 1 utilizando duas frequências distintas.

Uma abordagem bastante utilizada em modulação FSK é a que se mostra na figura 3.5, onde se varia a frequência portadora em função de um sinal digital binário modulante para gerar o sinal modulado. Nessa variação faz-se corresponder aos símbolos 0 e 1 as frequências do sinal modulado  $f_0 = f_p - \Delta f$  e  $f_1 = f_p + \Delta f$ , respectivamente, onde  $f_p$  é a frequência portadora e  $\Delta f$  o desvio de frequência utilizado.

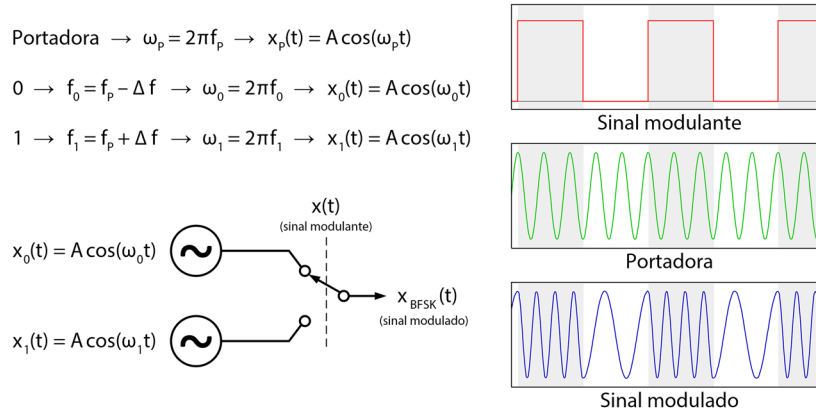


Figura 3.5. Modulador FSK tradicional. Baseado em [35].

Apesar da sua simplicidade, esta abordagem implicaria a utilização de dois osciladores que por sua vez aumentariam o consumo de energia e as dimensões do *transceiver*. Além disso, devido às várias comutações não seria garantida a continuidade da fase, o que consequentemente geraria componentes de frequência indesejáveis.

Como tal, optou-se por realizar a modulação utilizando os recursos já existentes no microcontrolador. Devido às características deste, a modulação terá de ser digital, pelo que surgem as hipóteses ASK, FSK e PSK. Como um dos *mixers* utilizados neste projecto, o NXP SA636, apenas desmodula directamente sinais ASK e FSK, a modulação PSK fica automaticamente excluída. Então, escolheu-se a modulação FSK por ser mais imune ao ruído e por ter apresentado bons resultados em [35] e [36]. A figura 3.6 apresenta o diagrama de blocos do modulador FSK implementado.

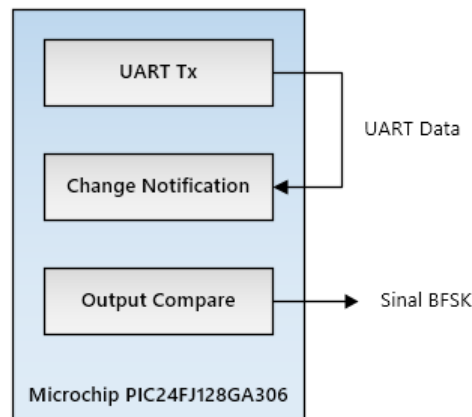


Figura 3.6. Diagrama de blocos do modulador FSK implementado.

Como se pode visualizar pela figura 3.6, para gerar o sinal BFSK são utilizados os seguintes módulos do microcontrolador:

- UART: cria as tramas UART.
- Change Notification: detecta as transições de nível ascendentes e descendentes do sinal da UART, isto é, as transições de  $0 \rightarrow 1$  ou de  $1 \rightarrow 0$  dos bits do sinal.
- Output Compare: gera o sinal BFSK quadrado.

Para gerar o sinal BFSK, o módulo UART primeiro cria as tramas e depois envia-as para o módulo Change Notification. Este por sua vez gera um pedido de interrupção no microcontrolador sempre que detecta uma mudança de estado nos bits do sinal da trama UART. Por fim, em cada pedido de interrupção que atende, a rotina de serviço à interrupção altera a frequência do sinal PWM (Pulse-Width Modulation) que é continuamente gerado pelo módulo Output Compare. Para garantir a continuidade de fase e desse modo permitir a conversão do sinal para uma sinusóide nos andares seguintes, o sinal PWM é gerado com *duty-cycle* de 50%. Na figura 3.7, ilustra-se um exemplo da modulação FSK implementada.

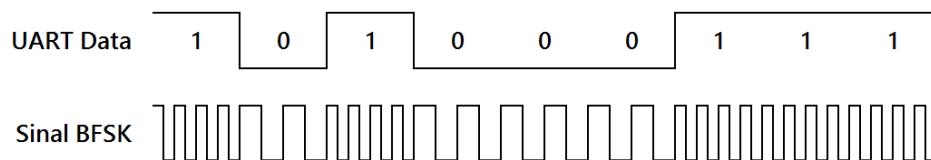


Figura 3.7. Exemplo da modulação FSK implementada.

O sinal BFSK gerado tem então as seguintes características:

- Forma de onda quadrada com *duty-cycle* de 50%
- Frequências: 666 KHz (0) e 800 KHz (1)
- Amplitude: 3 Vpp

Esta abordagem tem, no entanto, uma desvantagem. O facto de o sinal modulado ser quadrado em vez de sinusoidal implica que o mesmo tenha um espectro bastante disperso devido às harmónicas da sua frequência fundamental. Como tal, o sinal terá de ser posteriormente transladado na frequência e filtrado de modo a confinar o seu espectro e a permitir um tratamento mais eficiente. Apesar disso, tem as vantagens de consumir menos potência, de não aumentar as dimensões do *transceiver* e de garantir a continuidade de fase, como já foi referido.

### 3.5.2 IF up-conversion mixer

O modulador FSK não consegue gerar uma frequência elevada o suficiente para possibilitar uma transmissão eficiente do sinal através do corpo humano, pois como se viu anteriormente, o corpo humano apresenta a menor atenuação para a frequência de 10.7 MHz. Como tal, é necessário transladar o sinal modulado para esta frequência ou para uma frequência que por sua vez também apresente baixa atenuação.

Esta operação é possível utilizando um *mixer*, pelo que neste sistema foi utilizado o *mixer* NXP SA602A devido à sua simplicidade e ao seu baixo consumo. O seu consumo típico é de cerca de 2.4 mA quando alimentado a 5V. No entanto, como o *transceiver* é alimentado a 3V foi necessário adicionar a este último um multiplicador de tensão (*charge pump*) com saída regulada a 5V para alimentar o *mixer*. Para o efeito, a escolha recaiu sobre o Texas Instruments REG710, pois necessita de poucos componentes externos e para 5V consegue fornecer uma corrente de saída máxima de 30 mA, valor este mais que suficiente para alimentar os dois *mixers* NXP SA602A utilizados neste sistema de comunicação. Na secção 3.7 (Alimentação) apresenta-se a configuração utilizada para o multiplicador de tensão, o qual por sua vez também é utilizado para alimentar o *mixer* NXP SA602A do andar de RF *up-conversion* e o amplificador sintonizado do emissor. A figura 3.8 apresenta o diagrama de blocos do NXP SA602A.

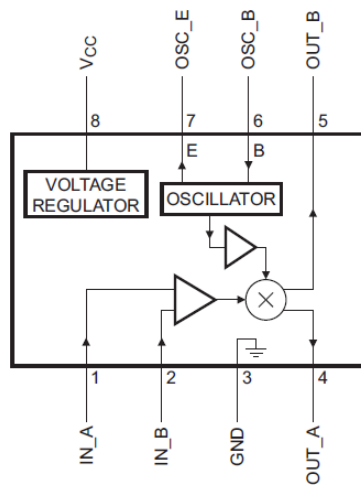


Figura 3.8. Diagrama de blocos do *mixer* NXP SA602A [49].

Como se pode ver pela figura 3.8, o NXP SA602A possui um regulador de tensão, um oscilador, um amplificador e tem a particularidade de poder ser configurado como *double-balanced* (apenas o produto das duas entradas é colocado na saída). No entanto, para simplificar a implementação do *transceiver*, optou-se por uma configuração não-balanceada. Devido ao alto ganho deste *mixer* (tipicamente 18 dB a uma frequência de 45 MHz), é necessário atenuar primeiro o sinal BFSK, pois este possui uma amplitude demasiado elevada (3 Vpp) para a sua entrada. Para tal, utilizou-se o circuito representado na figura 3.9, o qual é constituído por um divisor resistivo e por um filtro passa-banda, permitindo deste modo a atenuação do sinal e ao mesmo tempo a eliminação da componente DC e de outras componentes de frequência indesejadas. Verificou-se na prática que, depois de atenuado, o sinal BFSK passa a ter uma amplitude em torno dos 35 mVpp.

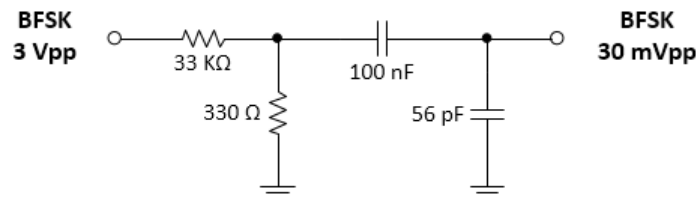


Figura 3.9. Atenuador e filtro passa-banda à entrada do *mixer* NXP SA602A.

Na figura 3.10 é apresentado o circuito que realiza a conversão do sinal BFSK para o sinal IF BFSK (sinal BFSK de frequência intermédia) e a posterior filtragem na banda de interesse.

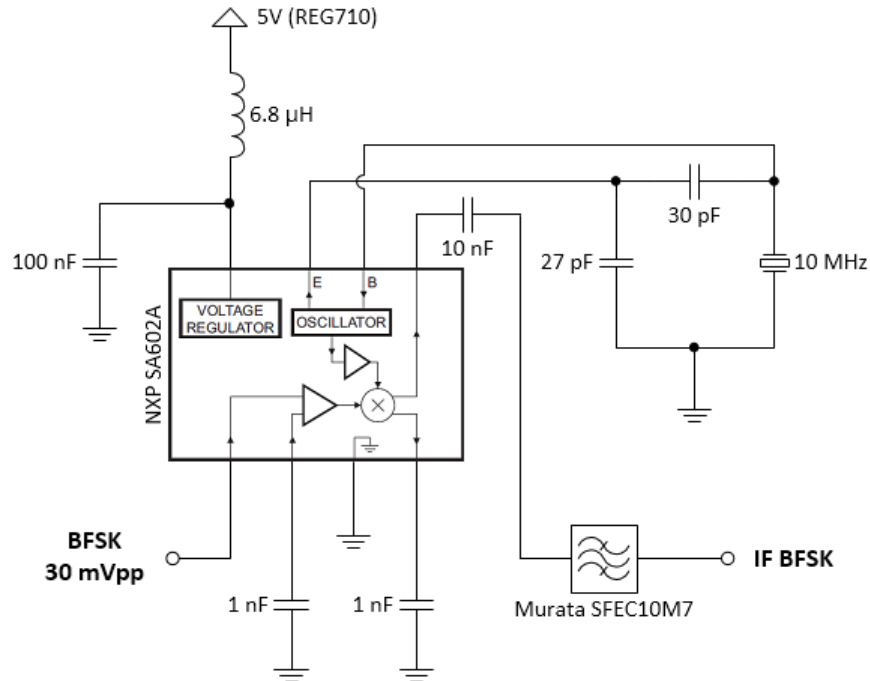


Figura 3.10. Circuito que realiza a conversão do sinal BFSK para o sinal IF BFSK e a posterior filtragem na banda de interesse.

Como se pode visualizar na figura 3.10, o circuito contém o *mixer* NXP SA602A e o filtro passa-banda cerâmico Murata SFEC10M7. O seu funcionamento do circuito será explicado de seguida.

Primeiro, o sinal BFSK já depois de ter sido atenuado é aplicado à entrada do *mixer* NXP SA602A, pelo que este último efectua a translacção do sinal para a frequência intermédia (FI) pretendida, como explica a figura 3.11.

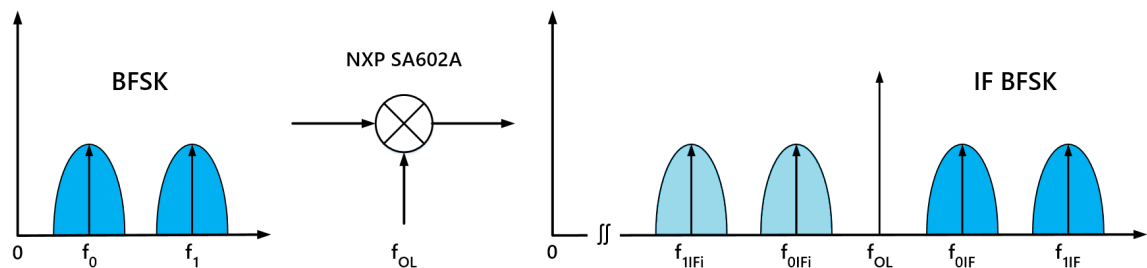


Figura 3.11. Processo de IF up-conversion efectuado pelo *mixer* NXP SA602A.

Neste processo o sinal BFSK e o sinal proveniente do oscilador local são multiplicados, resultando à saída do *mixer* um sinal com as seguintes componentes de frequência:

$$f_{0IF} = f_{OL} + f_0 \quad (3.1)$$

$$f_{1IF} = f_{OL} + f_1 \quad (3.2)$$

$$f_{0IFi} = f_{OL} - f_0 \quad (3.3)$$

$$f_{1IFi} = f_{OL} - f_1 \quad (3.4)$$

onde  $f_{OL}$  é a frequência de oscilação local e  $f_0$  e  $f_1$  as frequências correspondentes aos símbolos 0 e 1 do sinal BFSK, respectivamente.

Como se pode verificar pela figura 3.11, correspondeu-se o sinal IF BFSK à banda de frequências superior do sinal resultante à saída do *mixer*. Como tal, para o sinal IF BFSK ficar centrado na frequência intermédia de 10.7 MHz foi utilizado um oscilador local com frequência de 10 MHz, pelo que resultam as seguintes frequências para o sinal IF BFSK:

- $f_{0IF} = f_{OL} + f_0 = 10 \text{ MHz} + 666 \text{ KHz} = 10.666 \text{ MHz} \rightarrow \text{IF BFSK (0)}$
- $f_{1IF} = f_{OL} + f_1 = 10 \text{ MHz} + 800 \text{ KHz} = 10.800 \text{ MHz} \rightarrow \text{IF BFSK (1)}$

Portanto, as frequências  $f_{0IF}$  e  $f_{1IF}$  do sinal IF BFSK são as novas portadoras associadas aos símbolos 0 e 1, respectivamente.

No entanto, é necessário filtrar o sinal resultante à saída do *mixer* de modo a eliminar a banda de frequências inferior e todos os restantes harmónicos indesejados, restando apenas a banda superior, ou seja, o sinal IF BFSK. Esta é a função do filtro passa-banda Murata SFEC10M7 cuja resposta em frequência se apresenta na figura 3.12.

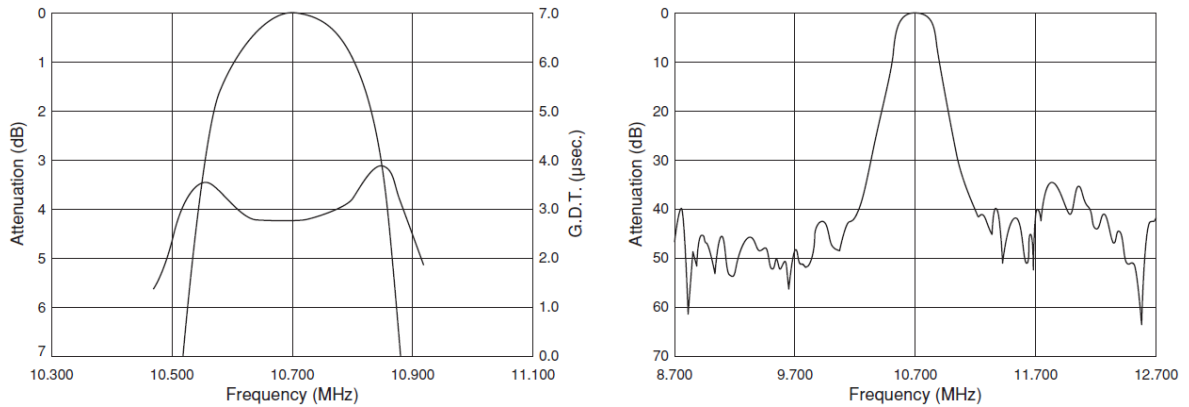


Figura 3.12. Resposta em frequência do filtro passa-banda Murata SFEC10M7 [50].

Este filtro tem uma banda de passagem (banda de frequências resultante da diferença entre as duas frequências às quais a atenuação é de 3 dB) de aproximadamente 280 KHz, centrada na frequência de 10.7 MHz. Fora da banda de passagem a atenuação é bastante elevada, chegando a atingir valores superiores a 30 dB. Como tal, este filtro torna-se bastante adequado para esta implementação, pois as frequências de interesse (10.666 MHz e 10.800 MHz) encontram-se dentro da sua banda de passagem. A figura 3.13 mostra o sinal resultante à saída do filtro passa-banda.

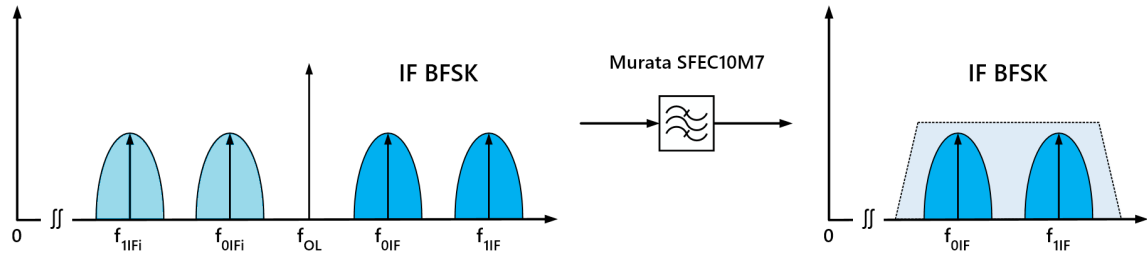


Figura 3.13. Sinal IF BFSK filtrado.

Como última nota refere-se o facto de que não foi efectuada a adaptação das impedâncias de saída do *mixer* e do filtro passa-banda, pois os filtros cerâmicos são pouco sensíveis a alterações de impedâncias de entrada/saída, o que permitiu poupar em número de componentes do *transceiver*. Além disto, foi necessária a utilização de condensadores de desacoplamento nas entradas e saídas do *mixer* não utilizadas devido ao facto de se utilizar uma configuração não-balanceada [36].

### 3.5.3 RF up-conversion mixer

Como se pode visualizar pela figura 3.14, o andar de RF *up-conversion* é semelhante ao IF *up-conversion*, apenas com a diferença de que não é efectuada uma filtragem após o processo de *up-conversion*. Neste andar utiliza-se novamente um *mixer* NXP SA602A para transladar o sinal IF BFSK para outra frequência de interesse (superior a 10.7 MHz), caso seja necessário. Por sua vez, a filtragem é efectuada no andar seguinte, isto é, no andar do amplificador sintonizado.

Verifica-se na prática que a amplitude do sinal IF BFSK à saída do filtro passa-banda situa-se em torno dos 15 mVpp, pelo que a mesma encontra-se abaixo do limite a aplicar à entrada do *mixer* NXP SA602A, não havendo por isso necessidade de atenuar o sinal novamente, ao contrário do andar anterior.

Para manter a frequência de transmissão de 10.7 MHz existem duas soluções. A primeira solução consiste em fazer o *bypass* deste andar, ligando a entrada directamente à saída. Por sua vez, a segunda consiste em injectar um sinal DC (0V) à entrada do oscilador local do *mixer* de modo a que o sinal de saída resulte praticamente numa réplica do sinal de entrada.

Apesar de a segunda solução degradar a relação sinal-ruído, verifica-se na prática que a degradação é aceitável, pelo que se optou por seguir por esta via. Para tal, simplesmente ligou-se as entradas do oscilador local directamente ao *ground* do circuito. Através desta solução demonstra-se também que é válida a utilização de um oscilador local variável para gerar à saída do *mixer* um sinal com a mesma frequência do sinal de entrada [36].

O processo de RF *up-conversion* deste andar é semelhante ao de IF *up-conversion* do andar anterior. O sinal IF BFSK é convertido no sinal RF BFSK através da translacção para a frequência pretendida, de modo semelhante ao que se ilustra na figura 3.11. Como se referiu acima, neste caso optou-se por manter a frequência de 10.7 MHz, pelo que o sinal RF BFSK é idêntico ao sinal IF BFSK.



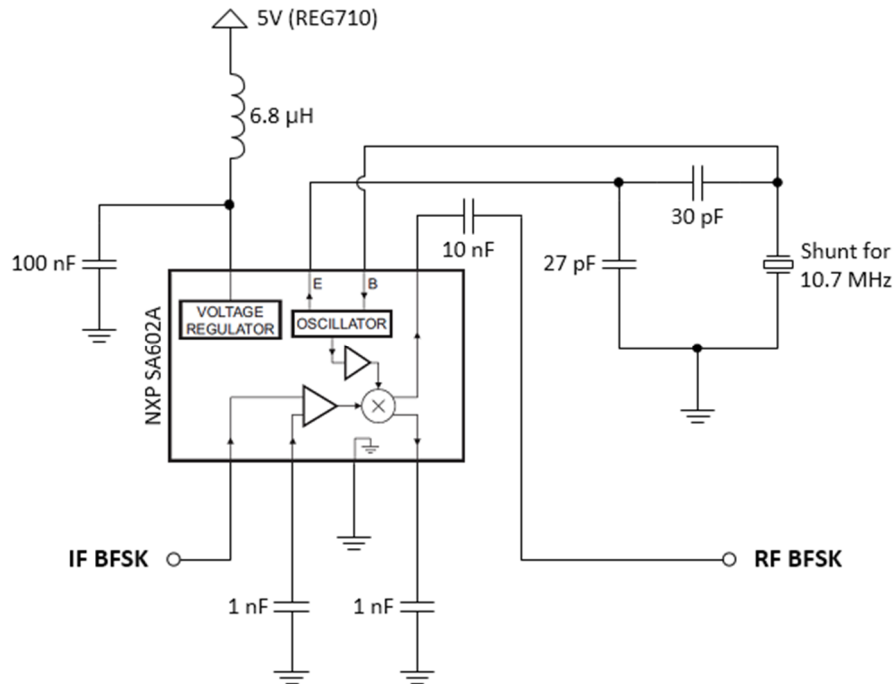


Figura 3.14. Circuito que realiza a conversão do sinal IF BFSK para o sinal RF BFSK.

Contudo, devido ao ruído que é adicionado ao sinal por consequência da solução escolhida, é necessário filtrar o sinal novamente de modo a obter apenas a banda de interesse, algo que será efectuado no andar seguinte, isto é, no amplificador sintonizado. Note-se que o *transceiver* implementado não é multicanal e, por isso, a filtragem do andar seguinte poderia ser descartada. No entanto, optou-se por não a descartar pois além de eliminar algum ruído adicional, poderá vir a ser útil no futuro no caso de ser implementado um *transceiver* com essa funcionalidade.

### 3.5.4 Amplificador sintonizado

O amplificador sintonizado é o circuito do andar de saída do emissor e tem como objectivo amplificar a gama das frequências de interesse, enquanto atenua todas as outras frequências. Na figura 3.15, apresenta-se o circuito do amplificador sintonizado.

Como se pode ver pela figura 3.15, o transistor escolhido para o amplificador foi o BJT NXP BFR520 por ter elevado ganho, baixo ruído, alta fiabilidade e elevada largura de banda. Observa-se também que este circuito utiliza uma malha LC ressonante como impedância de colector do transistor. Para a frequência de ressonância e suas frequências vizinhas a impedância desta malha torna-se bastante elevada, proporcionando assim o máximo de ganho ao amplificador apenas para a banda de interesse. No fundo, esta malha funciona como um filtro passa-banda para as frequências de interesse enquanto que para as restantes frequências a sua impedância torna-se praticamente um curto-circuito, acabando estas últimas por serem atenuadas ou pouco amplificadas. Os valores da bobine e do condensador foram determinados para uma frequência de ressonância de 10.7 MHz utilizando a expressão 3.5.

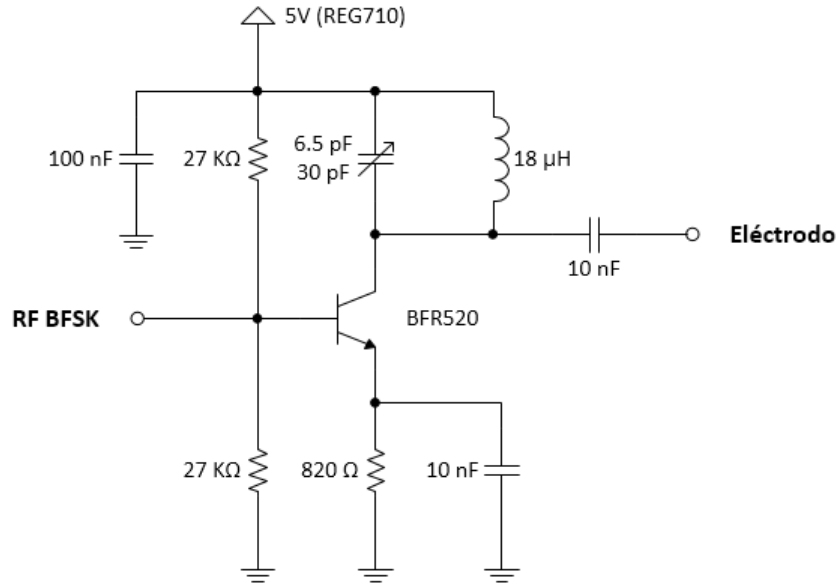


Figura 3.15. Amplificador sintonizado do andar de saída do emissor.

$$f_r = \frac{1}{2\pi\sqrt{LC}} \quad (3.5)$$

Este circuito tem a vantagem de ter menores perdas de energia e consequentemente uma menor tensão de alimentação precisamente por utilizar apenas componentes reactivos como impedância de colectador (malha LC), pois estes idealmente não dissipam potência, ao contrário dos componentes resistivos. Além disso permite selectividade, ou seja, permite amplificar apenas a gama de frequências pretendida através do ajuste do condensador da malha LC. Quanto maior for a selectividade, isto é, quanto maior for o factor de qualidade (Q) da malha LC, maior será a impedância desta à frequência de ressonância e nas suas proximidades e, portanto, mais estreita será a curva de ressonância da resposta em frequência da malha. Consequentemente, o ganho do amplificador sintonizado será maior para estas frequências, como ilustra a figura 3.16.

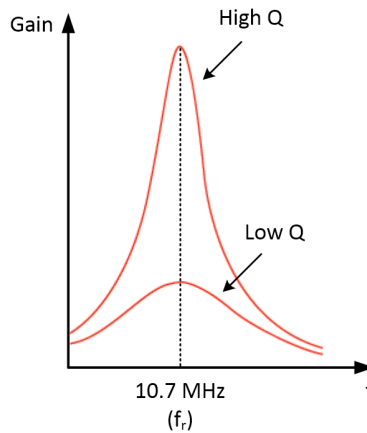


Figura 3.16. Ilustração da resposta em frequência do amplificador sintonizado do emissor consoante o factor de qualidade (Q) da malha LC.

As figuras 3.17 e 3.18 apresentam a resposta em frequência do amplificador sintonizado do andar de saída do emissor obtida por simulação em [35] e [36], respectivamente.

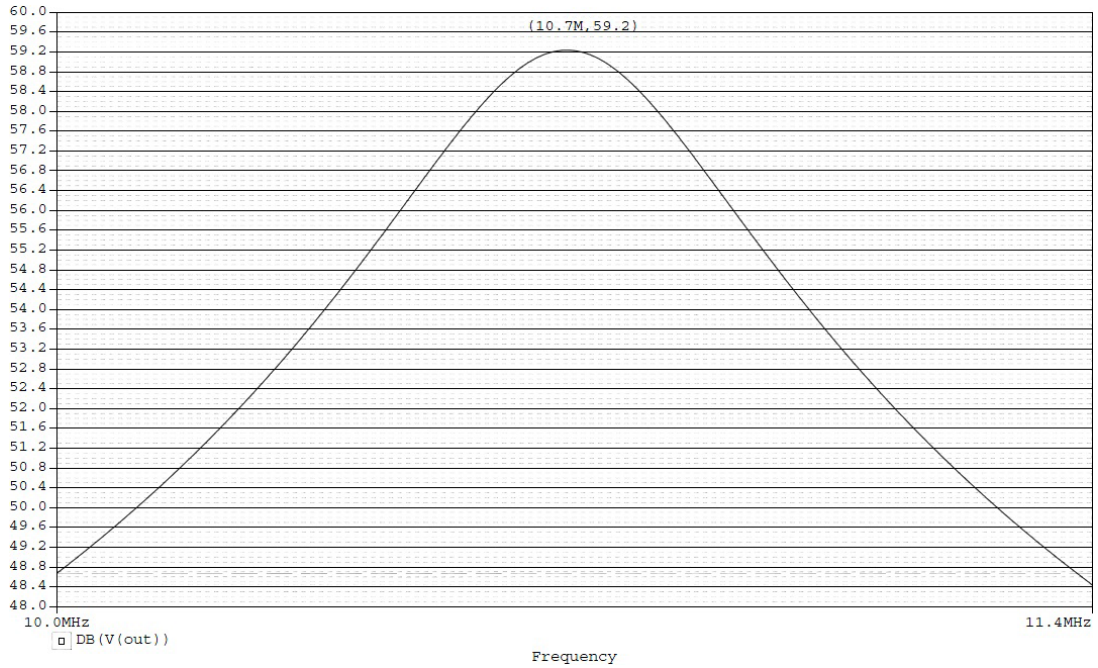


Figura 3.17. Resposta em frequência do amplificador sintonizado do emissor [35].

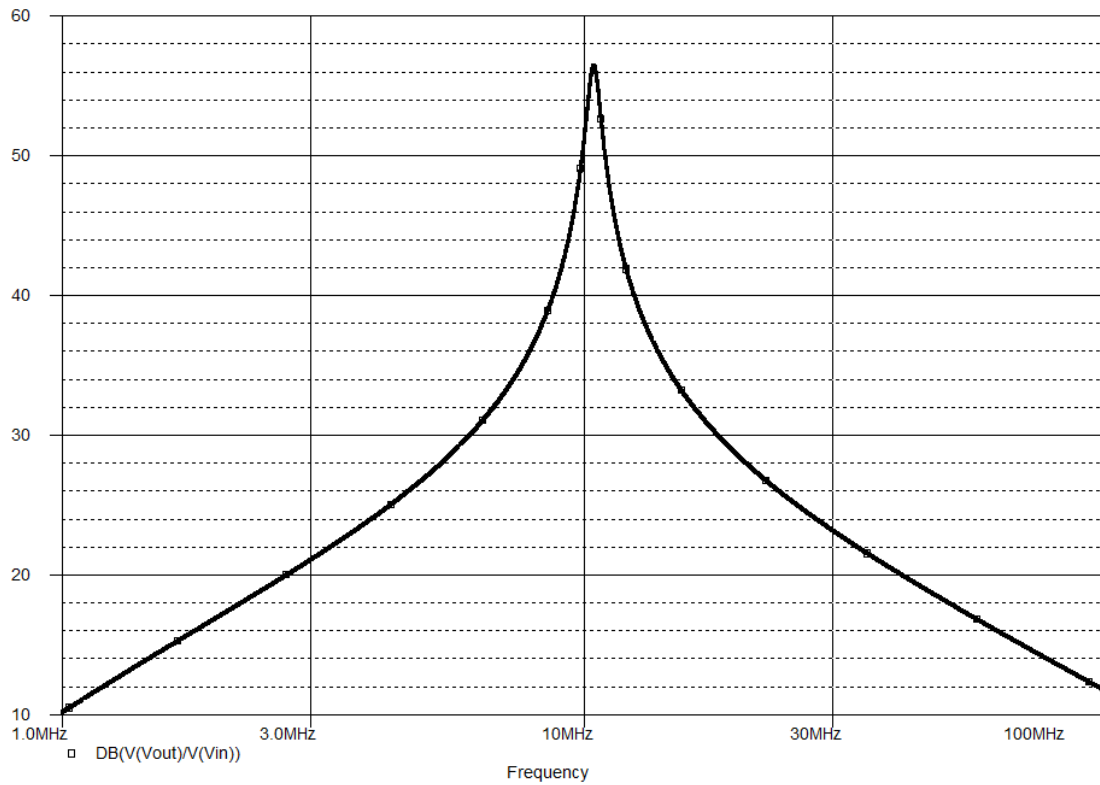


Figura 3.18. Resposta em frequência do amplificador sintonizado do emissor [36].

Tal como os *mixers* NXP SA602A, este circuito é também alimentado a 5V pelo *charge pump* Texas Instruments REG710. O eléctrodo do *transceiver* é partilhado por ambos os amplificadores sintonizados dos andares de entrada do receptor e saída do emissor, pelo que o facto de se usar o *charge pump* para alimentar este circuito permite activá-lo (polarizá-lo) apenas quando o *transceiver* se encontra no modo transmissor, impedindo desta forma o conflito (interferência) com o modo receptor. A activação ou desactivação do amplificador é efectuada através do microcontrolador, o qual por sua vez actua no pino de *enable* do *charge pump*.

## 3.6 Receptor

A figura 3.19 apresenta o diagrama de blocos do receptor.

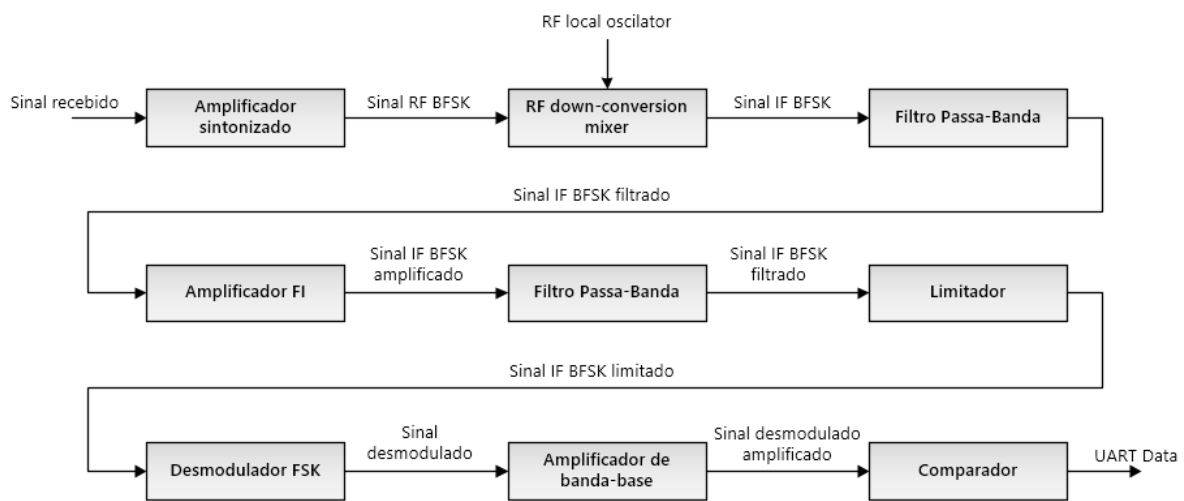


Figura 3.19. Diagrama de blocos do receptor.

Nas secções seguintes serão analisados em detalhe os vários blocos do receptor.

### 3.6.1 Amplificador sintonizado

Depois de transmitido, o sinal RF BFSK propaga-se pelo corpo humano até chegar ao andar de entrada do receptor. Este andar consiste num amplificador sintonizado que tem como principal objectivo operar como um *buffer* com uma capacidade de entrada bastante reduzida e uma resistência de entrada bastante elevada. Estas características são de extrema necessidade visto que a comunicação neste sistema é feita por acoplamento capacitivo. Tem também como objectivo amplificar a gama de frequências de interesse do sinal recebido, pois este sofre atenuação devido às próprias características do corpo humano. Contudo, a sua capacidade de amplificação é baixa, pelo que esta será realizada em grande parte no andar do amplificador FI. Na figura 3.20, apresenta-se o circuito do amplificador sintonizado.

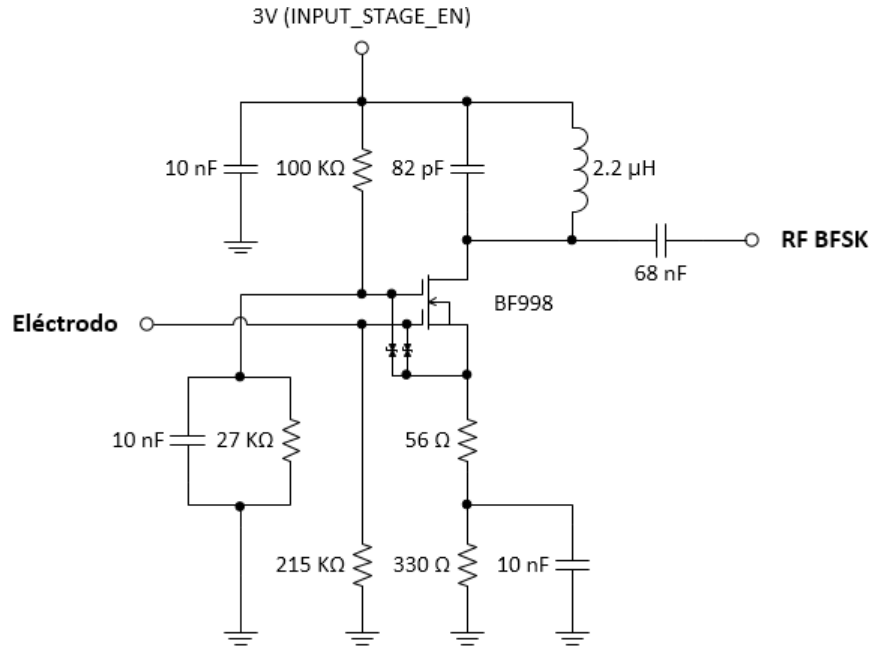


Figura 3.20. Amplificador sintonizado do andar de entrada do receptor.

Como se pode visualizar na figura 3.20, este amplificador utiliza um *dual-gate depletion* MOSFET NXP BF998 na configuração *cascode*, o qual é constituído por dois FETs onde um está configurado como fonte-comum (*common-source*) seguido por outro configurado como porta-comum (*common-gate*). Com esta configuração o efeito de Miller é diminuído, pois passa a não existir acoplamento directo entre a porta (*gate*) e o dreno (*drain*) do MOSFET inferior. Esta diminuição é imposta pela redução da resistência de entrada do andar superior e conseqüente redução do ganho de tensão do MOSFET inferior. O ganho do amplificador é depois recuperado pelo MOSFET superior. No entanto, este ganho não é elevado, razão pela qual a amplificação é depois realizada com maior ênfase no andar do amplificador FI. Desta configuração resulta também um aumento da largura de banda do amplificador, algo que pode ser útil no futuro se forem utilizadas outras frequências de trabalho para além da frequência de 10.7 MHz [35] e [36].

O motivo da escolha do MOSFET NXP BF998 deve-se à necessidade de este andar ter que apresentar uma capacidade de entrada bastante reduzida, como se mencionou anteriormente. O MOSFET escolhido apresenta uma capacidade de entrada de 2.1 pF, sendo por isso adequado para esta implementação.

De forma a economizar energia, este amplificador pode ser activado ou desactivado através de um pino proveniente do microcontrolador (INPUT\_STAGE\_EN), como mostra a figura 3.20. Ao ser activado, este pino alimenta o circuito com uma tensão igual à tensão de alimentação do microcontrolador, ou seja, 3V neste caso.

Para amplificar um pouco o sinal recebido e realizar alguma filtragem adicional, este amplificador utiliza uma malha LC de modo semelhante ao amplificador sintonizado do andar de saída do emissor. Como tal, o princípio de funcionamento é o mesmo, pelo que os valores da bobine e do condensador da malha foram determinados para a frequência de ressonância de 10.7 MHz, utilizando novamente a expressão 3.5. De igual modo, a relação entre o factor de qualidade da malha e o ganho do amplificador é semelhante, pelo que a ilustração da figura

3.16 também se aplica a este caso. As figuras 3.21 e 3.22 apresentam a resposta em frequência do amplificador sintonizado do andar de entrada do receptor obtida por simulação em [35] e [36], respectivamente.

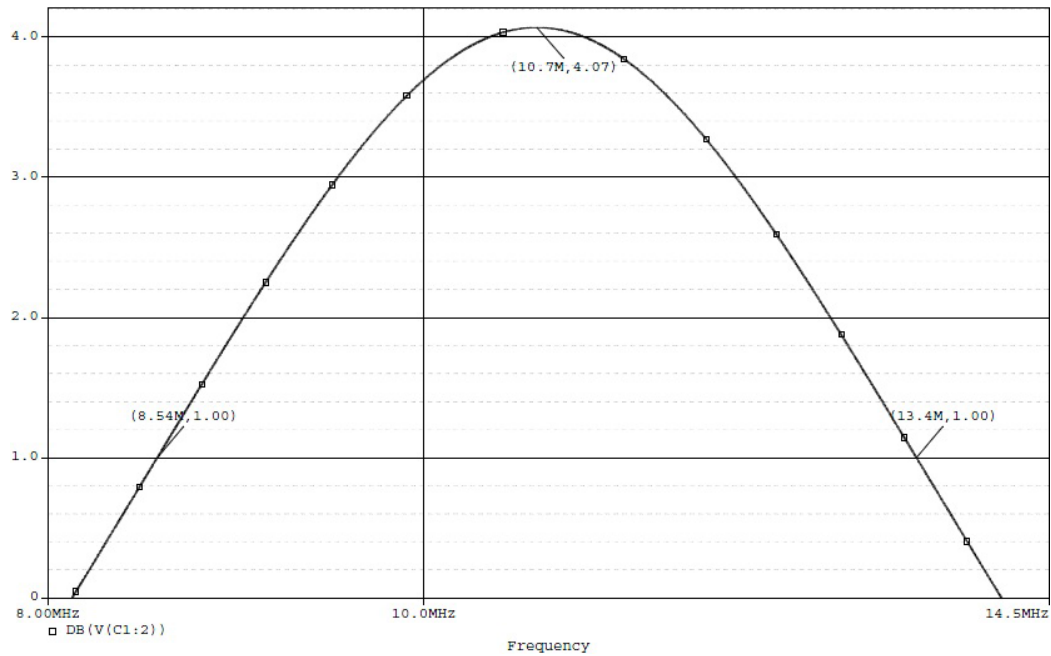


Figura 3.21. Resposta em frequência do amplificador sintonizado do receptor [35].

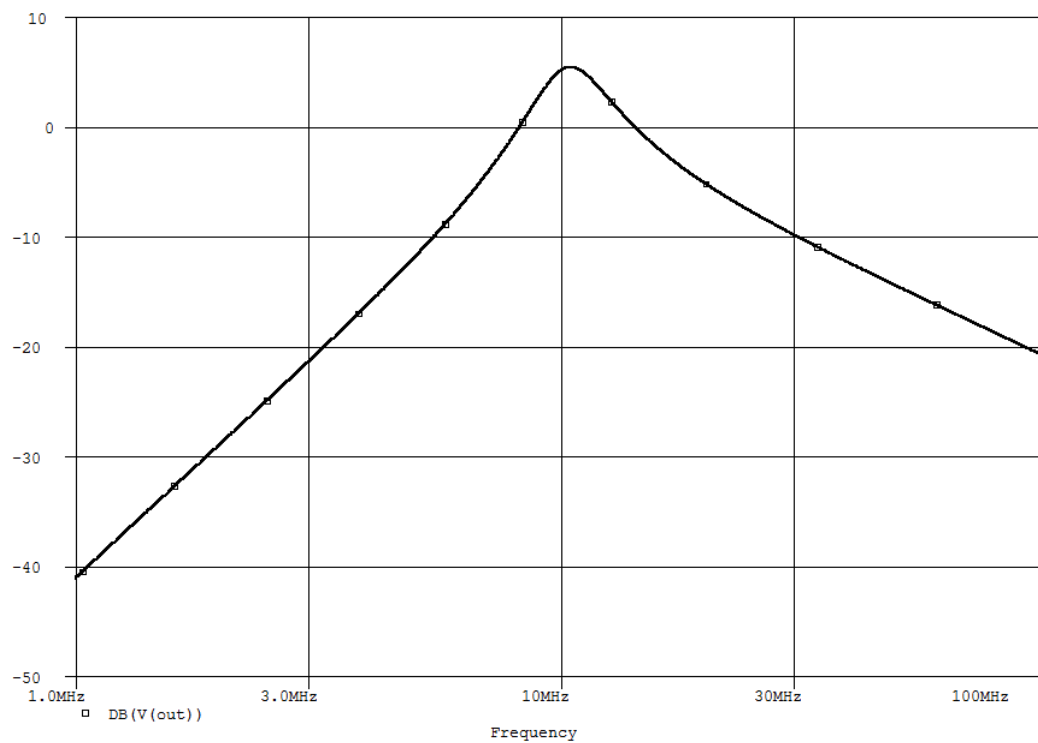


Figura 3.22. Resposta em frequência do amplificador sintonizado do receptor [36].

### 3.6.2 RF down-conversion mixer

Se durante o processo de RF *up-conversion* (realizado no emissor) o sinal for transladado para outra frequência intermédia diferente a 10.7 MHz, então na recepção o sinal presente à saída do amplificador sintonizado do receptor terá de ser transladado novamente para a frequência intermédia de 10.7 MHz de modo a ser tratado de forma mais conveniente. Esta é a função do bloco de RF *down-conversion*, no qual também é utilizado um *mixer*, tendo-se escolhido para o efeito o circuito integrado NXP SA636 principalmente devido ao seu baixo consumo (tipicamente 6.5 mA para uma tensão de 3V) e por possuir os seguintes módulos: *mixer*, oscilador, amplificador FI, limitador, detector de quadratura e uma saída RSSI (Received Signal Strength Indicator).

Como se verá nas secções seguintes, estes módulos serão de extrema utilidade no tratamento e desmodulação do sinal recebido. Relativamente à presente secção e como já foi mencionado, o módulo que será utilizado para realizar o processo de RF *down-conversion* é o *mixer*. Este tolera frequências de entrada até 1 GHz e possui um ganho tipicamente superior a 10 dB. Na figura 3.23 apresenta-se o diagrama de blocos do NXP SA636 onde se podem observar o *mixer* e os restantes módulos.

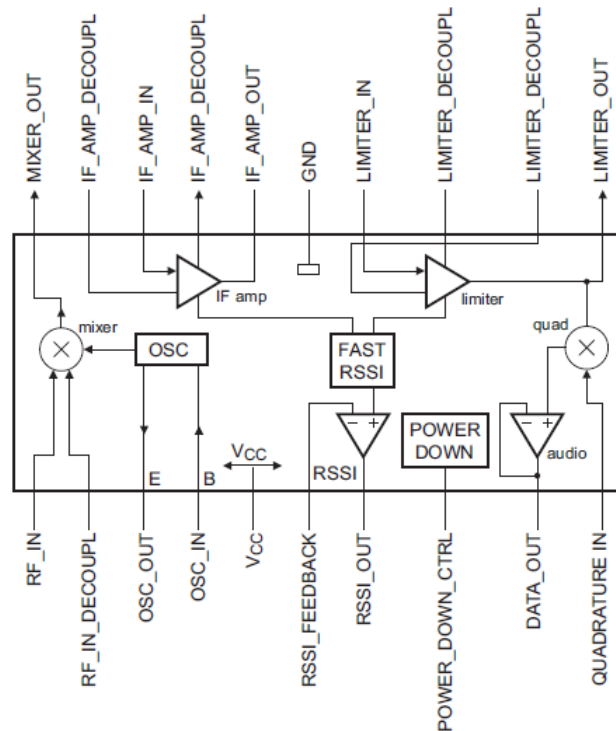


Figura 3.23. Diagrama de blocos do NXP SA636 [51].

O processo de RF *down-conversion* pode ser de dois tipos: *low-side* ou *high-side*. No caso *low-side*, a frequência do oscilador local é inferior à frequência intermédia do sinal RF BFSK. No caso *high-side*, a frequência do oscilador local é superior à frequência intermédia do sinal RF BFSK. Para estes dois casos resultam as seguintes componentes de frequência, respectivamente:

$$f_{IF_{low-side}} = f_{RF} - f_{OL} \quad (3.6)$$

$$f_{IF_{high-side}} = f_{OL} - f_{RF} \quad (3.7)$$

onde  $f_{IF}$  é a frequência intermédia do sinal IF BFSK resultante,  $f_{RF}$  a frequência intermédia do sinal RF BFSK e  $f_{OL}$  a frequência do oscilador local.

Para explicar este processo irão fazer-se algumas considerações e suposições. Em primeiro lugar, considera-se que o sinal RF BFSK corresponde apenas às componentes de frequência resultantes da soma da frequência do oscilador local do andar de RF *up-conversion* com as frequências do sinal IF BFSK do andar de IF *up-conversion* ( $f_{0IF}$  e  $f_{1IF}$ ). Deste modo, a frequência intermédia do sinal RF BFSK será sempre superior a 10.7 MHz (o que não é verdade pois na realidade decidiu manter-se a frequência intermédia de 10.7 MHz para o sinal RF BFSK). Considera-se ainda o caso *low-side* (por ser mais fácil de ilustrar) e supõe-se que os amplificadores sintonizados dos andares de saída do emissor e entrada do receptor não atenuam o sinal RF BFSK para frequências acima de 10.7 MHz (o que também não é verdade pelas razões já mencionadas anteriormente). A figura 3.24 procura explicar de forma simples o processo de RF *down-conversion*. Mais uma vez, note-se que neste sistema de comunicação o sinal IF BFSK resultante deverá ficar centrado em 10.7 MHz.

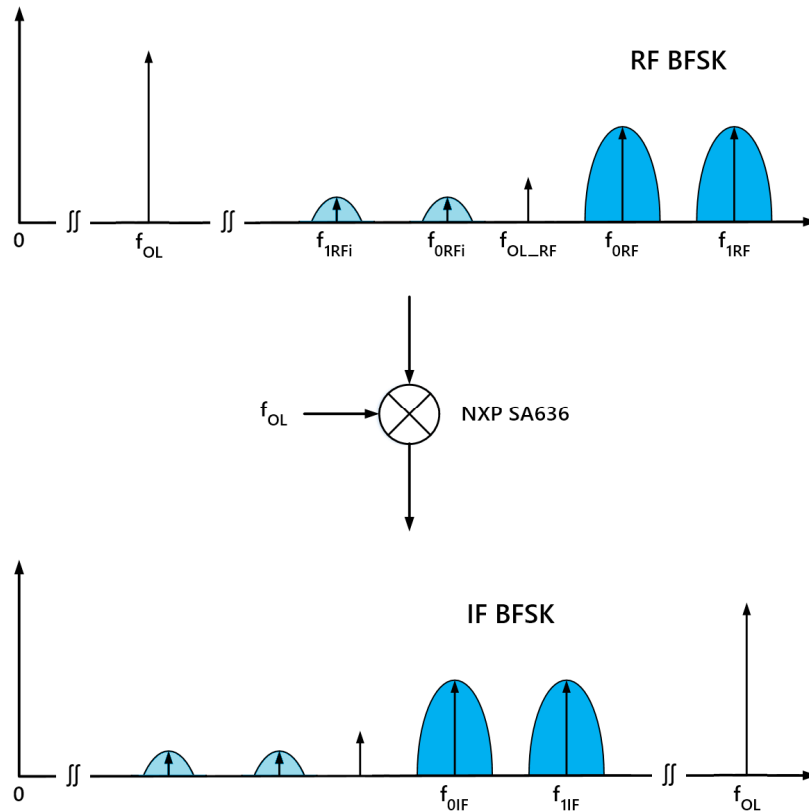


Figura 3.24. Processo de RF *down-conversion* (caso *low-side*).

A implementação do processo de RF *down-conversion* é ainda hipotética para o *transceiver* actual, servindo apenas como base para eventuais implementações futuras de um *transcei-*



ver multicanal. Assim, ao se ter decidido manter a frequência intermédia de 10.7 MHz no andar de RF *up-conversion*, como consequência não existe a necessidade de se realizar o processo de RF *down-conversion*. Então, tal como para o andar de RF *up-conversion*, neste andar também se ligou as entradas do oscilador local directamente ao *ground* do circuito.

Como se pode observar pela figura 3.24, o sinal resultante terá de ser filtrado antes de ser aplicado ao amplificador FI, pois surgem naturalmente componentes de frequência indesejadas devido aos processos de RF *conversion*. Para filtrar o sinal é utilizado novamente um filtro passa-banda Murata SFECF10M7. Na figura 3.25, apresenta-se o circuito do andar de frequência intermédia (FI) que é utilizado nos vários processos de tratamento e desmodulação do sinal. Além da presente secção, este será utilizado nas secções 3.6.3 (Amplificador FI), 3.6.4 (Limitador) e 3.6.5 (Desmodulador FSK).

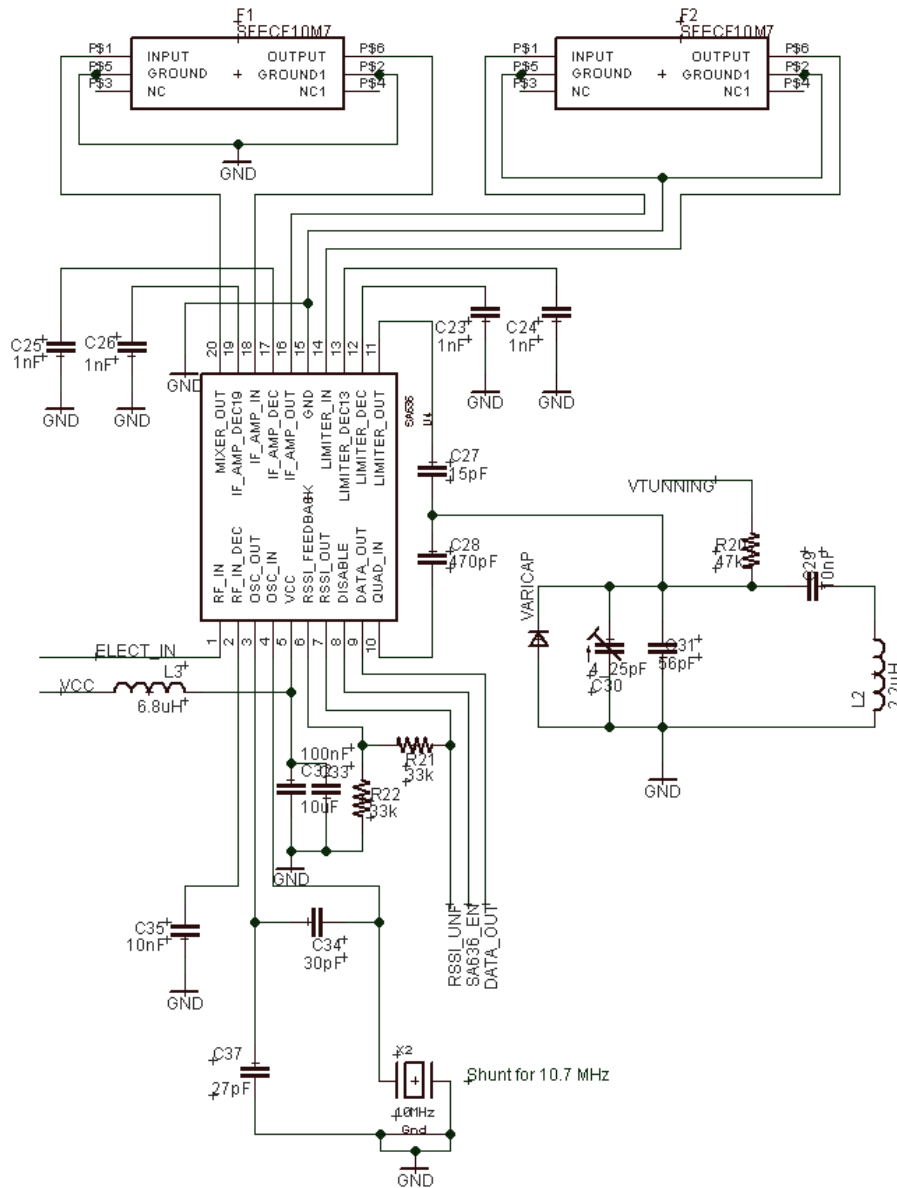


Figura 3.25. Circuito do andar de frequência intermédia (FI) do receptor.

### 3.6.3 Amplificador FI

O sinal transmitido sofre sempre alguma atenuação no corpo humano, como já foi referido. Anteriormente, verificou-se que o amplificador sintonizado do andar de entrada do receptor tem como função principal operar apenas como *buffer* de entrada, não apresentando por isso grande capacidade de amplificação. Como tal, o sinal recebido não possui amplitude suficiente para ser tratado e desmodulado devidamente, pelo que é necessário proceder à sua amplificação. Esta é a função do amplificador FI (amplificador de frequência intermédia) do NXP SA636, o qual em conjunto com o limitador consegue proporcionar um ganho de 92 dB ao sinal aplicado na sua entrada. Depois de amplificado, o sinal à sua saída terá de ser filtrado novamente, pois as componentes de frequência fora da banda de interesse são também amplificadas. Para tal é usado outro filtro passa-banda Murata SFEC10M7, como pode ser observado na figura 3.25.

### 3.6.4 Limitador

O limitador tem como função limitar a amplitude do sinal de modo a que a informação seja mantida apenas na modulação em frequência. O limitador utilizado encontra-se no NXP SA636 e permite limitar sinais com largura de banda até 25 MHz. Este amplifica o sinal de forma a que o mesmo sature, reduzindo deste modo a modulação em amplitude (AM) e o ruído causados durante a transmissão. Assim, a amplitude do sinal à entrada do detector de quadratura (desmodulador FSK) torna-se constante para qualquer que seja a distância entre o emissor e o receptor, permitindo desta forma que a modulação em frequência seja detectada com maior eficiência.

### 3.6.5 Desmodulador FSK

O sinal à saída do limitador ainda se encontra modulado em frequência e, como tal, é necessário proceder à sua desmodulação. O desmodulador FSK utilizado neste sistema de comunicação consiste no detector de quadratura presente no NXP SA636.

Para desmodular o sinal recebido este converte as variações de frequência em variações de fase e depois apresenta uma saída directamente proporcional a essas mesmas variações de fase. Na realidade, o detector de quadratura utilizado é constituído por um detector de fase (*mixer*) interno ao NXP SA636 e um circuito tanque ressonante externo ao NXP SA636. Na figura 3.26, apresenta-se o aspecto básico de um circuito detector de quadratura e, na figura 3.27, o detector de quadratura utilizado neste sistema de comunicação.

Como se pode visualizar na figura 3.26, o sinal modulado é dividido em dois e um deles é aplicado directamente ao detector de fase enquanto o outro é aplicado a um condensador de elevada reactância sofrendo assim um desfasamento de 90 graus. De seguida, este último é aplicado a um circuito tanque ressonante sintonizado à frequência central do sinal modulado, o qual tem como função criar desvios de fase proporcionais ao desvio da frequência do sinal relativamente à frequência de ressonância do circuito. Por fim, é também aplicado ao detector de fase. A diferença de fase entre os dois sinais é então igual a 90 graus (provocados pelo condensador de elevada reactância) mais o desfasamento provocado pelo circuito ressonante. O detector de fase apresenta depois à sua saída uma tensão directamente proporcional a esta diferença, a qual permite obter o sinal em banda-base. No caso particular da modulação BFSK

existem duas portadoras iguais à frequência central mais ou menos um desvio de frequência, pelo que são estes desvios de frequência que provocam os avanços ou atrasos de fase adicionais por via do circuito ressonante [35] [36].

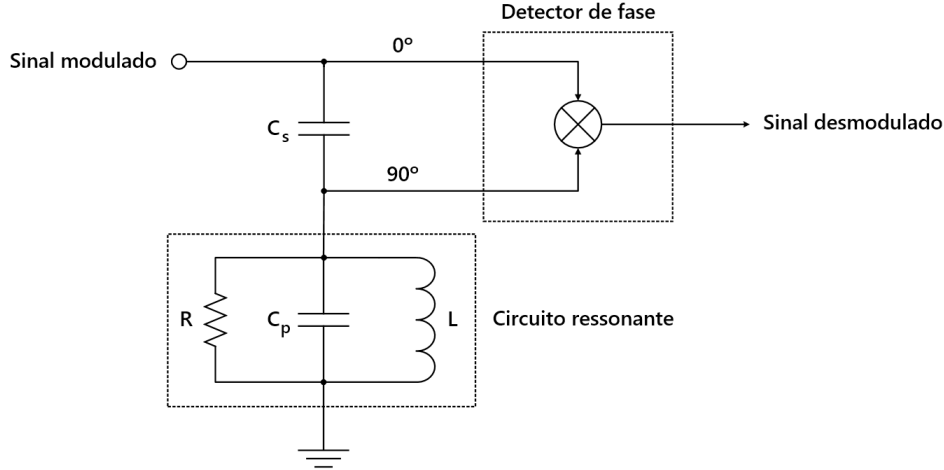


Figura 3.26. Aspecto básico de um circuito detector de quadratura. Baseado em [35].

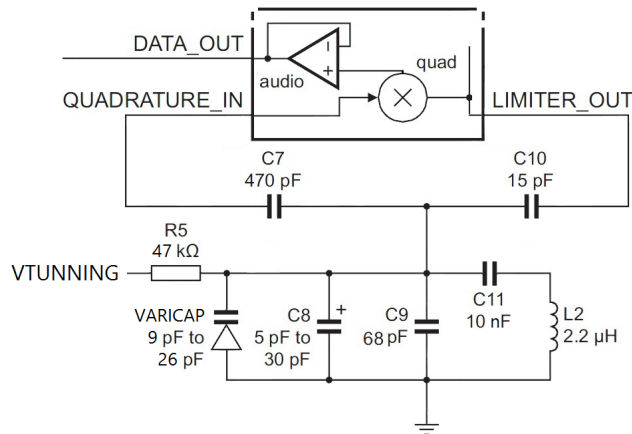


Figura 3.27. Detector de quadratura utilizado. Baseado em [51].

As deduções matemáticas e derivações do detector de quadratura representado na figura 3.26 foram explicadas detalhadamente em [35], pelo que não se justifica a repetição desse trabalho nesta dissertação, dada a sua complexidade. Apenas se refere que a variação de fase é tanto maior quanto maior for o factor de qualidade ( $Q$ ) do circuito ressonante e que existe um compromisso entre este último e o desvio de frequência relativamente à frequência de ressonância utilizada. Para um elevado factor de qualidade existe uma maior excursão de sinal mas o desvio de frequência máximo admissível para o sinal torna-se mais reduzido. Por outro lado, para um baixo factor de qualidade, apesar de a excursão do sinal ser menor este apresenta maior linearidade para maiores desvios de frequência.

Relativamente ao detector de quadratura utilizado neste sistema de comunicação, pela figura 3.27 é possível observar que o condensador C7 é o que provoca o desfaseamento de 90 graus e que o circuito ressonante é composto pelos condensadores C8, C9, C10 e C11, pela bobine L2, pela resistência R5 e pelo *varicap*. O condensador C11 serve apenas para separar

a tensão de referência do *varicap* da tensão da bobine L2, pelo que se escolheu um valor baixo (10 nF) para não influenciar o comportamento do circuito ressonante. O circuito ressonante é sintonizado para a frequência de 10.7 MHz através da variação da capacidade do *varicap*, pois este tem a vantagem de permitir sintonias finas por ser mais preciso do que um condensador variável. O *varicap* escolhido foi o Skyworks Solutions SMV1256, pois permite grandes variações de capacidade para diferenças de tensão até 3V [36].

A tensão de referência do *varicap* é controlada pelo microcontrolador, o qual gera um sinal PWM (com frequência de 500 KHz) que por sua vez é aplicado a um filtro passa-baixo (circuito RC com frequência de corte de 10 KHz). Daqui resulta um sinal de controlo (tensão de referência) praticamente DC (sinal VTUNNING da figura 3.27). É então através da variação do *duty-cycle* do sinal PWM que é possível aumentar ou diminuir a tensão de referência do *varicap*. As figuras 3.28 e 3.29 ilustram o processo de geração da tensão de referência do *varicap* e a sua variação com o *duty-cycle* do sinal PWM gerado pelo microcontrolador, respectivamente.

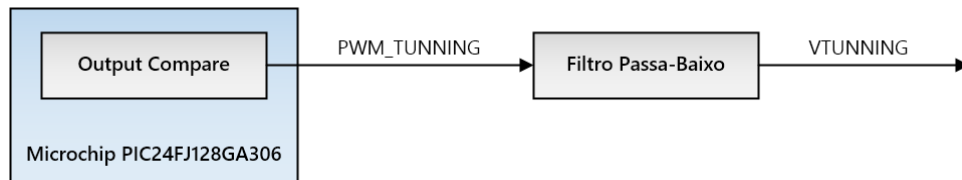


Figura 3.28. Geração da tensão de referência do varicap (sinal VTUNNING).

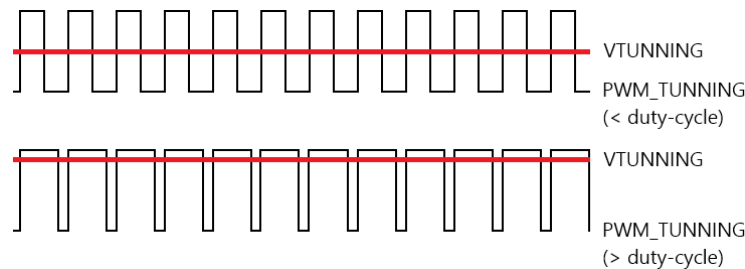


Figura 3.29. Ilustração da variação da tensão de referência do varicap consoante a variação do *duty-cycle* do sinal PWM gerado pelo microcontrolador.

Na figura 3.30, apresenta-se uma simulação da resposta em frequência do detector de quadratura utilizado.

Pela figura 3.30, conclui-se que à frequência de ressonância ideal (10.7 MHz) existe um desfasamento de cerca de 78 graus entre o sinal aplicado directamente ao detector de fase (*mixer*) e o sinal aplicado ao circuito ressonante. Apesar de este desfasamento não coincidir com o desfasamento teórico de 90 graus, os resultados práticos mostram que o detector de quadratura consegue desmodular correctamente o sinal IF BFSK. A discrepância verificada relativamente ao desfasamento deve-se ao facto de o circuito ressonante não ter uma sintonia perfeita, ou seja, a frequência de ressonância deste não iguala a frequência portadora do sinal recebido (10.7 MHz). Pelos gráficos da figura 3.30, é possível ainda observar que são as variações de frequência em torno da frequência de ressonância que provocam as variações de fase do sinal de forma mais acentuada. Os resultados práticos demonstram também que o sinal desmodulado, isto é, o sinal à saída detector de quadratura, possui uma tensão com amplitude máxima de cerca de 150 mVpp.

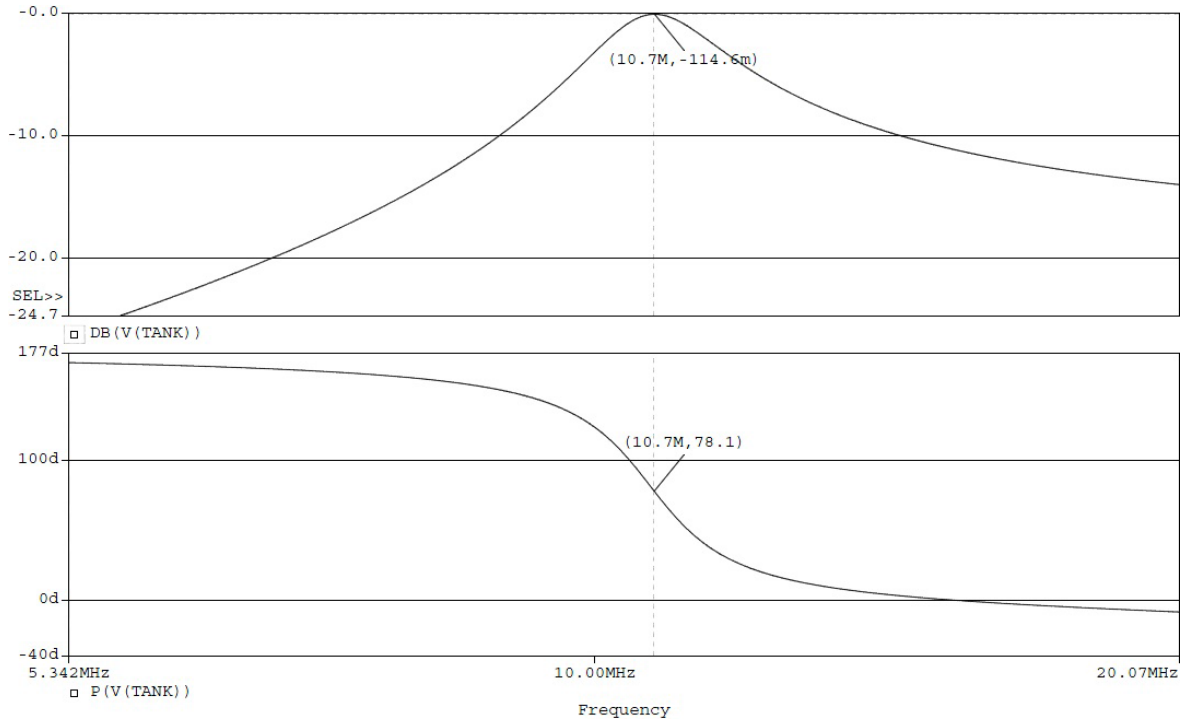


Figura 3.30. Simulação da resposta em frequência do detector de quadratura utilizado [35].

Verifica-se também que o sinal desmodulado possui uma componente DC que por sua vez varia consoante a sintonia do circuito ressonante do detector de quadratura. A sua existência deve-se mais uma vez ao facto de o circuito ressonante não ter uma sintonia perfeita. Esta ocorrência pode ser explicada da forma que se apresenta de seguida [36]. Primeiro que tudo, é conhecido o facto de que o produto interno entre dois sinais ortogonais é um sinal nulo. Por outras palavras, o produto interno entre dois sinais desfasados de 90 graus é igual a 0, como mostra a expressão 3.8:

$$A \cos(\omega t) \cos(\omega t + 90) = 0 \quad (3.8)$$

Este seria o caso de uma sintonia perfeita do circuito ressonante, pois o sinal aplicado a este iria estar desfasado exactamente 90 graus do sinal aplicado directamente ao detector de fase (*mixer*), pelo que do produto destes sinais, isto é, do sinal à saída do *mixer* resultaria uma componente DC nula. No caso de uma sintonia imperfeita (pode afirmar-se que ocorre sempre no mundo real), o sinal aplicado ao circuito ressonante não está desfasado exactamente 90 graus do sinal aplicado directamente ao detector de fase (*mixer*) e, como tal, à saída deste último resulta um sinal com uma componente DC diferente de 0, como mostra a expressão 3.9:

$$A \cos(\omega t) \cos(\omega t + 90 + \alpha) = \frac{A}{2} (\cos(2\omega t + 90 + \alpha) + \cos(90 + \alpha)) \neq 0 \quad (3.9)$$

Além disto, verifica-se que neste caso o NXP SA636 adiciona também uma tensão de *offset* DC ao sinal desmodulado, pelo que é necessário sintonizar o circuito ressonante tendo também em conta este factor.

### 3.6.6 Amplificador de banda-base

O sinal desmodulado possui uma amplitude muito reduzida e por isso é necessário amplificá-lo antes de o aplicar ao circuito de decisão do receptor. Para tal recorre-se a um amplificador, mais especificamente, a um *opamp* em configuração não inversora.

O *opamp* utilizado é o Texas Instruments LMH6647, pois além de ser *rail-to-rail*, apresenta um baixo consumo (650  $\mu\text{A}$  quando activo e 50  $\mu\text{A}$  quando inactivo), baixa tensão de alimentação (2.5V no mínimo) e largura de banda adequada para esta implementação (55 MHz). Na figura 3.31, apresenta-se o circuito do amplificador de banda-base.

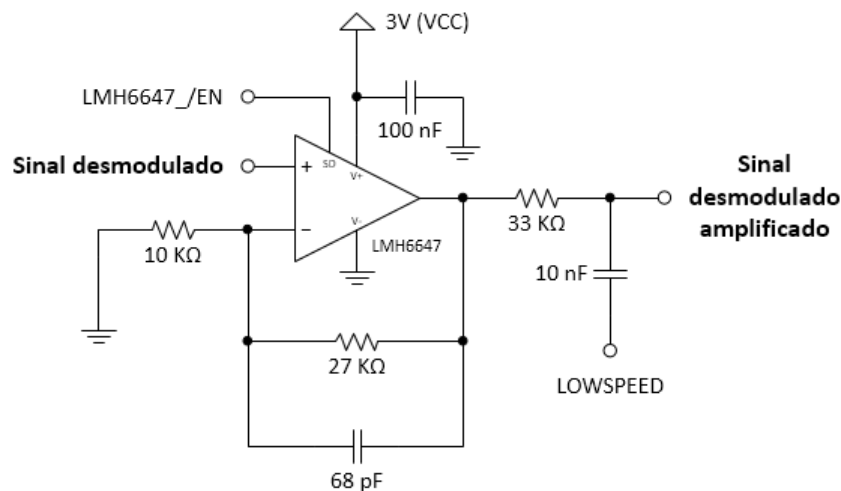


Figura 3.31. Circuito do amplificador de banda-base.

O sinal à saída do desmodulador FSK possui uma componente DC que ao ser amplificada pode saturar o sinal à saída do amplificador de banda-base. Então, o ganho deste amplificador foi determinado de modo a que a componente DC do sinal nunca ultrapasse o limiar de decisão máximo do comparador do andar seguinte cujo valor é de 2.15V, neste caso. O ganho determinado foi de 2.1, obtendo-se assim um sinal à saída do amplificador com uma tensão de amplitude máxima de cerca de 315 mVpp.

Como se pode observar no circuito da figura 3.31, são ligados ao amplificador dois pinos do microcontrolador. Um é o pino LMH6647\_/EN, o qual serve para activar ou desactivar o *opamp*, permitindo assim uma maior economização de energia. O outro é o pino LOWSPEED, o qual serve para activar ou desactivar o modo de recepção lenta (*low-speed*) do *transceiver*. Este modo serve para diminuir a taxa de transmissão para 500 bps no caso de se verificar que o canal de transmissão é bastante ruidoso ou que a distância entre o emissor e o receptor é relativamente longa. Para tal é utilizado um filtro passa-baixo à saída do *opamp* (circuito RC com frequência de corte de aproximadamente 480 Hz) cuja massa se encontra ligada ao pino de LOWSPEED. Para activar o modo, o pino LOWSPEED é configurado como porto digital e é-lhe atribuído o valor 0 (ficando com uma tensão de 0V), pelo que o filtro passa-baixo passa a ficar activo e é então filtrado o ruído de alta frequência do sinal à saída do *opamp*. Contudo, como já foi mencionado, sacrifica-se a taxa de transmissão pois neste caso passam apenas as baixas frequências. Para desactivar o modo, o pino LOWSPEED é configurado como porto analógico e consequentemente passa a apresentar uma impedância bastante elevada, tendo como resultado a desactivação do filtro passa-baixo.

### 3.6.7 Comparador

De modo a tornar possível a leitura do sinal recebido por parte do microcontrolador é necessário converter o sinal desmodulado num sinal digital binário com limites superior e inferior bem definidos. Para tal é utilizado o circuito de decisão que se apresenta na figura 3.32.

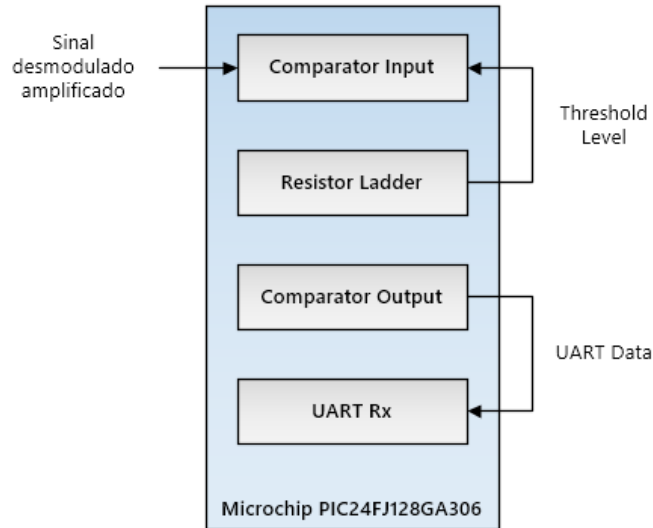


Figura 3.32. Diagrama de blocos do circuito de decisão utilizado para a conversão do sinal desmodulado num sinal digital binário.

Como se pode observar pela figura 3.32, este circuito é constituído por um comparador, por um gerador de tensões de referência (*resistor ladder*) e pelo módulo UART, todos eles internos ao microcontrolador. Ao se optar por esta solução poupa-se mais uma vez no consumo de energia e nas dimensões do *transceiver*. Contudo, o facto de não se utilizar um comparador externo impossibilita a implementação de um comparador com histerese, o qual minimizaria a ocorrência de erros devido às perturbações causadas pelo ruído.

O sinal desmodulado é aplicado à entrada do comparador e à saída deste obtém-se o sinal digital a ser encaminhado para o módulo UART. O comparador compara o sinal de entrada com uma tensão de referência e coloca à sua saída um sinal com apenas duas tensões possíveis, isto é, um sinal com tensão igual à tensão de saturação superior ( $V_{CC}$ ) no caso em que o sinal de entrada é superior à tensão de referência e um sinal com tensão igual à tensão de saturação inferior ( $V_{SS}$ ) no caso em que o sinal de entrada é inferior à tensão de referência. A saída do comparador pode ser considerada digital precisamente por apresentar apenas duas tensões, 0V ou 3V, as quais correspondem aos símbolos 0 e 1 lógicos, respectivamente. Na figura 3.33, apresenta-se o diagrama de blocos do módulo de triplo comparador do microcontrolador.

O gerador de tensões de referência consiste numa escada de resistências que permite gerar tensões de referência programáveis entre 0V e 2.15V, neste caso. A tensão de referência do comparador é programada tendo em conta o nível médio de tensão do sinal desmodulado e é encaminhada para a entrada do comparador, definindo assim o limiar de decisão deste relativamente ao sinal. Esta solução permite obter a informação do sinal nos casos em que o *offset* é maior do que o suposto devido a uma sintonia incorrecta do circuito ressonante do detector de quadratura. O diagrama de blocos do gerador de tensões de referência pode ser visualizado na figura 3.34.

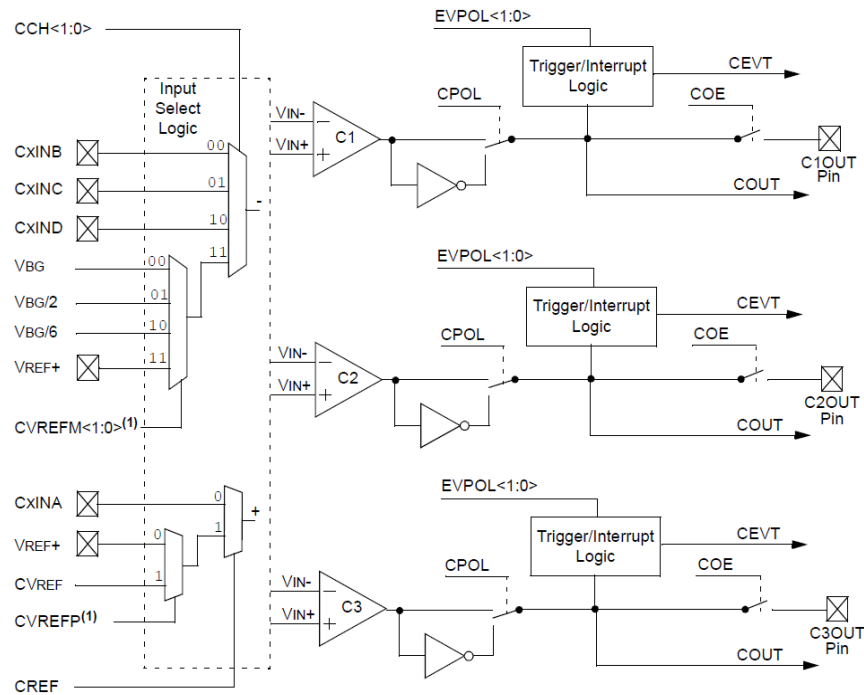


Figura 3.33. Diagrama de blocos do módulo de triplo comparador do microcontrolador [52].

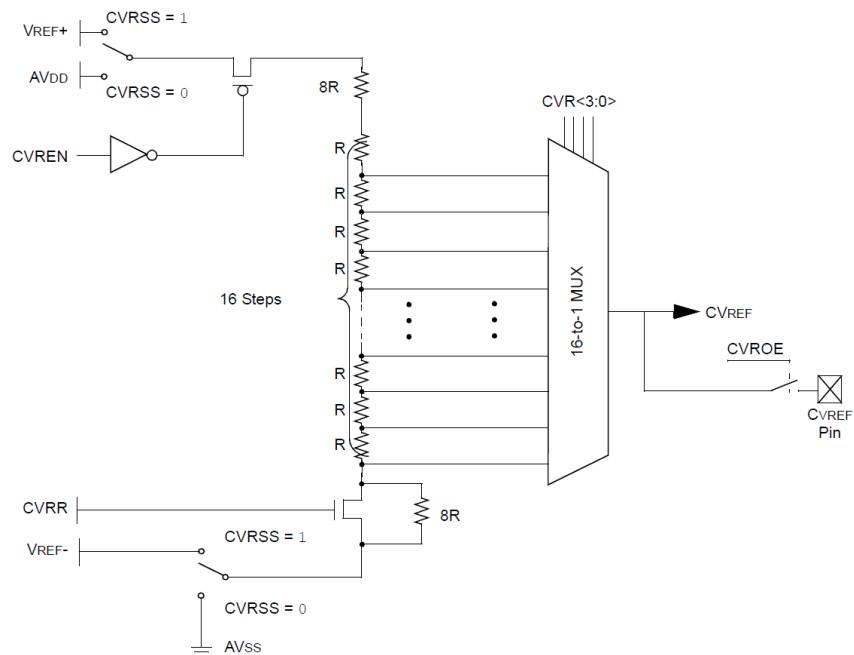


Figura 3.34. Diagrama de blocos do gerador de tensões de referência (*resistor ladder*) do microcontrolador [52].

Outra solução para gerar a tensão de referência consiste no uso de um filtro passa-baixo que permita gerar uma tensão contínua a partir do sinal desmodulado, obtendo-se assim um nível médio que varia dinamicamente consoante o *offset* do sinal.



## 3.7 Alimentação

### 3.7.1 Regulador de tensão

O *transceiver* implementado é alimentado por uma pilha recarregável do tipo botão (LIR2032) seguida de um regulador de tensão com saída regulada de 3V. Escolheu-se o regulador Texas Instruments TLV70030 devido ao facto de este apresentar uma baixa tensão de *dropout*, baixo ruído à saída, reduzida corrente quiescente e uma excelente performance transiente. É utilizada a configuração apresentada na figura 3.35 de forma a reduzir o ruído e o *ripple* à saída do regulador, garantindo assim maior estabilidade e uma boa resposta transiente.

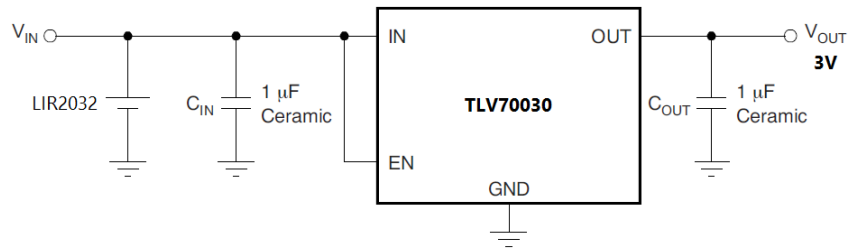


Figura 3.35. Configuração utilizada para o Texas Instruments TLV70030. Baseado em [53].

O regulador foi projectado para fornecer à sua saída uma tensão regulada de 3V e uma corrente máxima de 200 mA.

### 3.7.2 Controlador de gestão de carga

Implementou-se um circuito que permite carregar a pilha através de uma ligação micro-USB. Para tal, utiliza-se um conector micro-USB seguido de um controlador de gestão de carga, o qual fornece à sua saída a potência necessária para carregar a pilha. O controlador escolhido foi o Microchip MCP73831 devido às suas reduzidas dimensões e à sua alta precisão na regulação da tensão de predefinição. Este permite programar uma corrente de carga entre 15 mA e 500 mA e possui 4 opções de tensão regulada, sendo elas 4.2V, 4.35V, 4.4V e 4.5V. Além disso, possui um modo de *power-down* automático, o que permite economizar energia sem serem necessárias quaisquer implementações adicionais. Na figura 3.36 apresenta-se o circuito do carregador micro-USB do *transceiver*, no qual se pode visualizar a configuração utilizada para o controlador de gestão de carga.

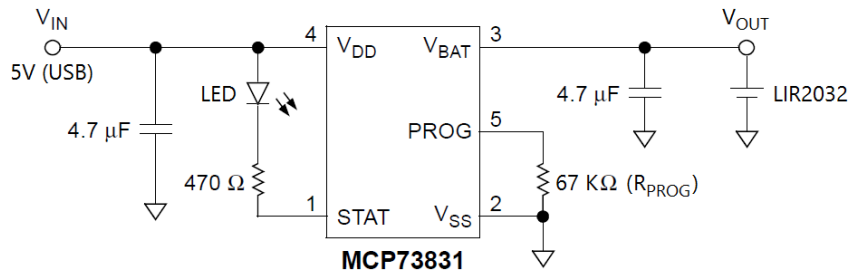


Figura 3.36. Configuração utilizada para o Microchip MCP73831. Baseado em [54].

### 3.7.3 Multiplicador de tensão

A figura 3.37 apresenta a configuração utilizada para o multiplicador de tensão (*charge pump*) que alimenta os *mixers* NXP SA602A dos andares de IF e RF *up-conversion* e o amplificador sintonizado do andar de saída do emissor. Com esta configuração é possível obter o máximo de eficiência e o mínimo de *ripple* à saída se  $C_{IN}$ ,  $C_{PUMP}$  e  $C_{OUT}$  forem condensadores SMD cerâmicos, recomendação esta que se teve em conta no projecto.

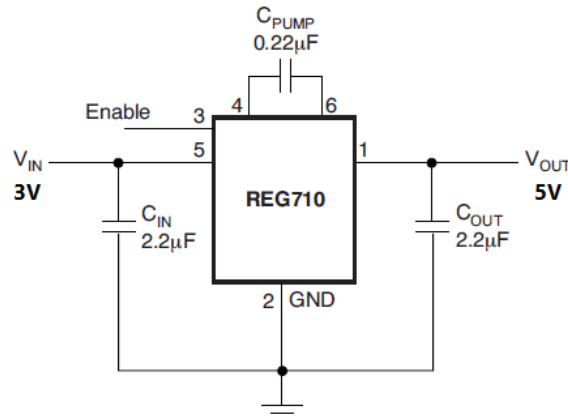


Figura 3.37. Configuração utilizada para o Texas Instruments REG710 [55].

Tal como se mencionou na secção 3.5.2, o *charge pump* escolhido foi o Texas Instruments REG710, pois necessita de poucos componentes externos, apresenta uma tensão regulada de 5V à sua saída e fornece uma corrente de saída máxima de 30 mA.

## 3.8 Placa de circuito impresso

Tomando novamente como base o trabalho realizado em [36], neste projecto também se procurou desenvolver uma placa de circuito impresso (PCB) de tamanho reduzido e de depuração simples. De igual modo, procurou-se reduzir ao máximo o efeito do elevado ganho dos andares de amplificação do NXP SA636 pois estes podem originar oscilações e consequente instabilidade no receptor por via dos habitualmente desprezáveis fenómenos de realimentação (*feedback*) que derivam do *layout* do circuito. Além disto, o acoplamento capacitivo entre as pistas é outro factor que pode originar fenómenos de realimentação [36]. Como tal, para o desenvolvimento da PCB seguiram-se as seguintes directrizes:

- *Layout* de 4 camadas (*layers*) com planos de massa sólidos para reduzir o acoplamento capacitivo entre as pistas e aumentar o desempenho global do sistema.
- Construção do tipo SMT (SMD) e utilização de componentes com *packages* do tipo SOIC e SSOP de tamanho reduzido.
- Condensadores de desacoplamento o mais perto possível do NXP SA636 para se reduzirem os efeitos provenientes dos elementos parasitas.
- Pistas não perpendiculares de modo a evitar radiações indesejadas provenientes das descontinuidades das pistas.

- Vias em todos os componentes com ligações à massa.
- Inclusão de diversos pontos de teste para fácil medição dos sinais relevantes.

A par destas implementações foram também substituídos todos os cristais da versão anterior do *transceiver* por outros de baixo consumo e tamanho reduzido. Os novos cristais utilizados são os da série Abracon ABM8. Foram ainda incluídas duas portas micro-USB, uma para a programação do microcontrolador e outra para a alimentação do *transceiver* e comunicação de dados UART. Apesar do esforço realizado no sentido de se reduzir ao máximo o tamanho da PCB, o facto de existirem diversos pontos de teste acaba por restringir a sua miniaturização. No entanto, conseguiu desenvolver-se uma PCB menor que as das suas versões anteriores, com dimensões de 5 x 5 cm.

A figura 3.38 apresenta as 4 camadas da PCB desenvolvida. Para o seu desenvolvimento foi utilizado o programa Eagle 6.5 Light. No final, a sua produção ficou ao encargo de uma empresa externa à Universidade de Aveiro.

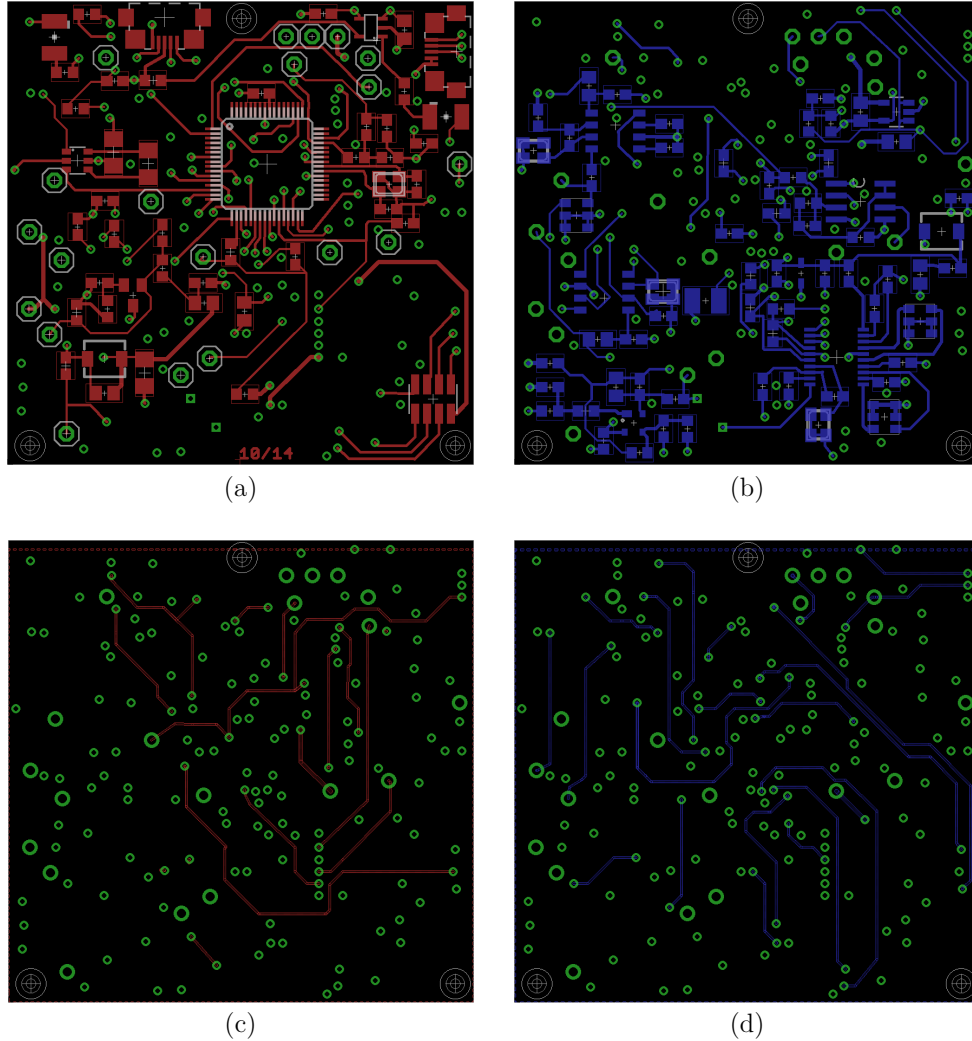


Figura 3.38. PCB desenvolvida. (a) Camada superior (*top layer*). (b) Camada inferior (*bottom layer*). (c) Camada superior intermédia. (d) Camada inferior intermédia.

### 3.9 Considerações finais

Como se referiu anteriormente, o *transceiver* implementado no âmbito deste projecto de dissertação foi fortemente baseado no da sua versão anterior, tendo-se efectuado melhorias principalmente a nível da redução do seu consumo de energia e do seu tamanho.

Uma das suas principais modificações foi o desenvolvimento de uma PCB de 4 camadas, a qual permitiu diminuir o acomplamento capacitivo entre as pistas, melhorar o desempenho global do sistema e reduzir o tamanho do *transceiver*. Outra foi a inclusão de um novo microcontrolador e a realização do seu respectivo remapeamento e reprogramação relativamente ao microcontrolador anterior, algo que permitiu diminuir o consumo de energia e adicionar novas funcionalidades ao *transceiver*. Convém referir que a substituição dos cristais foi outro factor que também contribuiu para a redução do tamanho e do consumo do *transceiver*.

Por sua vez, a inclusão das duas portas micro-USB permitiu uniformizar e simplificar a programação e comunicação UART com o microcontrolador, além de permitir alimentar o *transceiver* e carregar a pilha deste por USB em conjunto com o controlador de gestão de carga também introduzido nesta nova versão.

Foram ainda substituídas as resistências de *feedback* do amplificador de banda-base do receptor por outras de diferente valor de forma a impedir que o sinal desmodulado saturasse à sua saída e que o seu nível médio se encontrasse acima do limiar de decisão do comparador do microcontrolador. Foi feito também um ajuste ao filtro passa-baixo que gera este nível médio, tendo sido substituídos os componentes que o constituem por outros de diferente valor de modo a diminuir a sua frequência de corte e consequentemente a gerar um nível médio com uma tensão praticamente contínua. Ao se utilizar o nível médio do sinal desmodulado como tensão de referência de comparação, obtém-se assim um nível médio de referência que varia dinamicamente com os níveis superior e inferior do sinal desmodulado. Esta solução permite compensar algum eventual *offset* do sinal e ainda assim obter os níveis de decisão correctos após a descodificação do mesmo.

## Capítulo 4

# Sistema de pagamento

### 4.1 Sumário

De forma semelhante ao capítulo anterior, neste capítulo descreve-se em detalhe o sistema de pagamento implementado no âmbito desta dissertação de mestrado.

Primeiro é apresentada a arquitectura do sistema de pagamento, a qual procura explicar como este foi concebido na sua globalidade e como o mesmo se interliga ao sistema BCC. Depois são apresentadas todas as implementações realizadas, nomeadamente a interface Android que efectua a interligação do sistema de pagamento ao sistema BCC e todo o *hardware* e *software* desenvolvido nesse sentido. No final são mencionadas algumas considerações relativamente à implementação e à segurança do sistema de pagamento desenvolvido.

### 4.2 Conceito e arquitectura do sistema de pagamento

Como se referiu anteriormente, esta dissertação de mestrado tem como objectivo principal o desenvolvimento de um sistema de pagamento seguro usando comunicação através do corpo humano. Após ponderação e estudo de várias possibilidades, chegou-se a um conceito final para o sistema a desenvolver, o qual se descreverá de seguida.

O sistema de pagamento implementado consiste na comunicação entre um *smartphone* e um terminal de pagamentos onde o canal de transmissão é logicamente o corpo humano. A comunicação entre os dois dispositivos é efectuada utilizando o sistema BCC implementado onde um dos *transceivers* se encontra interligado ao *smartphone* e o outro ao terminal de pagamentos. O *smartphone* pertencerá a um utilizador (cliente) e deverá ser encarado como a sua carteira digital. Por sua vez, o terminal de pagamentos pertencerá a uma entidade ou indivíduo que comercialize bens e/ou serviços. A figura 4.1 ilustra o sistema de pagamento concebido no âmbito deste projecto, ao qual se deu a designação de sistema de pagamento via BCC (ou *contactless* BCC).

Para efectuar um pagamento, o utilizador pode deixar o seu *smartphone* no bolso ou em qualquer outra zona próxima do seu corpo, pois todo o processo é automático, pelo que o utilizador necessita apenas de se encontrar também ele próximo do terminal de pagamentos até ser notificado acerca do sucesso da operação. O utilizador apenas terá de tocar no terminal de pagamentos quando for solicitada a sua autenticação (inserindo o seu código pessoal ou a sua impressão digital) e a confirmação do pagamento, garantindo-se assim uma maior segurança.

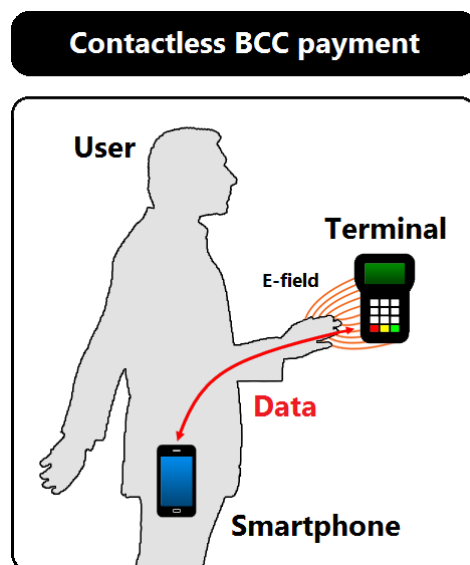


Figura 4.1. Ilustração do sistema de pagamento concebido.

Devido à dificuldade em se obter um terminal de pagamentos físico e que possuisse portas de comunicação I/O, optou-se por simular o mesmo através de outro *smartphone*. Os *smartphones* estão equipados com o sistema operativo Android, o qual é um sistema operativo móvel baseado no *kernel* do Linux que disponibiliza ferramentas para estabelecer a interface com unidades exteriores via USB. Este é desenvolvido pela Open Handset Alliance (OHA), liderada pela Google, e actualmente é de longe o sistema operativo móvel mais utilizado a nível mundial, como mostra a figura 4.2. Este foi um dos principais motivos pela sua escolha para este projecto, além do facto de ser um sistema operativo bastante versátil e flexível, podendo correr num elevado número de *smartphones* de diferentes marcas e modelos.

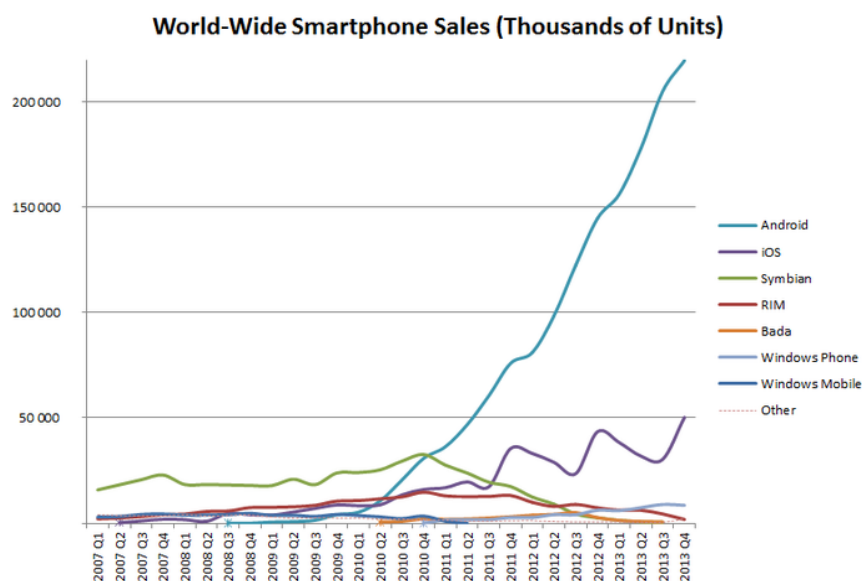


Figura 4.2. Número de *smartphones* vendidos em todo o mundo nos últimos anos para diversos sistemas operativos móveis [56].

Ambos os *smartphones* do utilizador e do terminal emulado correm a sua aplicação/serviço Android, as quais comunicam entre si e contêm todo o código necessário ao processamento das diversas fases do processo de pagamento.

Para tornar isto possível é necessário implementar uma interface de interligação entre os *smartphones* e os *transceivers* do sistema BCC. De modo a simplificar a sua implementação e a evitar soluções que certamente conduziram a desastres (danificação de *smartphones* por exemplo), optou-se por efectuar a interligação recorrendo a uma placa de desenvolvimento especialmente concebida para sistemas e protótipos onde se pretenda a comunicação entre *smartphones* equipados com Android e *hardware* externo. A placa de desenvolvimento utilizada é a IOIO-OTG para Android, apresentada na figura 4.3. Em nota de curiosidade, esta placa foi inventada no ano de 2010 por Ytai Ben-Tsvi (actualmente a trabalhar como engenheiro de *software* na Google em Sunnyvale, Califórnia, EUA) quando este se deparava com enormes dificuldades em interligar *smartphones* a motores externos que permitiriam controlar pequenos *robots*. Com o decorrer do tempo e fruto do trabalho de diversos colaboradores, a placa foi sempre evoluindo, pelo que tem sido continuamente melhorada e actualizada até aos dias de hoje tanto a nível de *hardware* como de *firmware*.

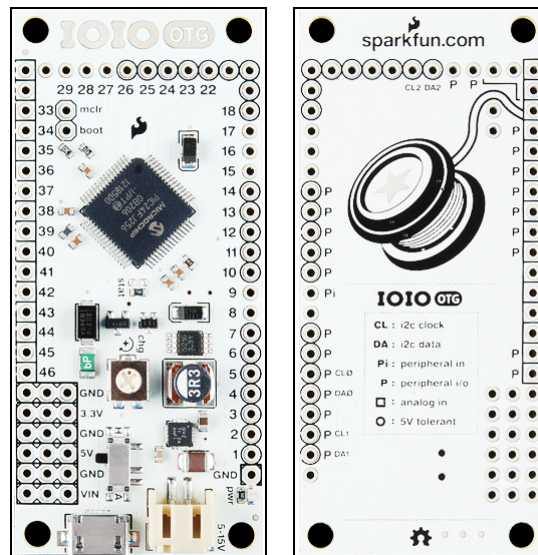


Figura 4.3. Placa de desenvolvimento IOIO-OTG.

A IOIO-OTG é essencialmente uma placa de desenvolvimento que contém um microcontrolador (Microchip PIC24FJ256GB206) e pinos para a efectuar a interligação entre um dispositivo Android e algum tipo de dispositivo/*hardware* externo. A mesma é suportada por *firmware* e bibliotecas desenvolvidas para Android no sentido de minimizar e simplificar o trabalho necessário para a implementação de *hardware/software* que permita a comunicação entre os dispositivos. Deste modo, a IOIO-OTG permite que a aplicação Android desenvolvida pelo programador se torne independente da camada física utilizada, pelo que o controlo e a gestão da mesma será efectuada por uma Java API de alto-nível. A comunicação entre os dispositivos pode ser realizada utilizando módulos como GPIO, PWM, ADC, I<sup>2</sup>C, SPI ou UART.

Esta placa inclui uma porta micro-USB para efectuar a ligação a um *smartphone* ou PC e um conector JST para alimentação externa. Tem ainda a particularidade de ser uma placa de pequenas dimensões, com cerca de 7 cm de comprimento e 3 cm de largura. O facto de ser

OTG representa uma vantagem relativamente às suas primeiras versões pois funciona segundo as especificações do USB On-The-Go, podendo assim conectar-se ou como *host* (*master*) ou como *device/acessory* (*slave*) a um dispositivo Android.

Desta forma, durante o processo de pagamento o *smartphone* do utilizador e o terminal de pagamentos (simulado por outro *smartphone*) trocam informação entre si através das aplicações Android e da interface de interligação (na qual se utilizam duas placas IOIO-OTG) em conjunto com os *transceivers* do sistema BCC.

Na figura 4.4, apresenta-se a arquitectura simplificada do sistema de pagamento implementado, a qual procura resumir tudo o que foi mencionado até este ponto.

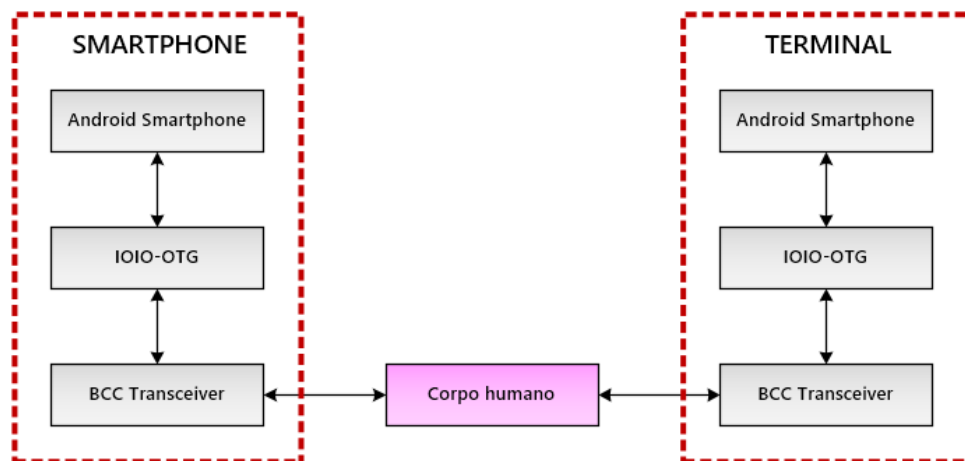


Figura 4.4. Arquitectura simplificada do sistema de pagamento implementado.

Na figura 4.5, apresenta-se a arquitectura detalhada do sistema de pagamento.

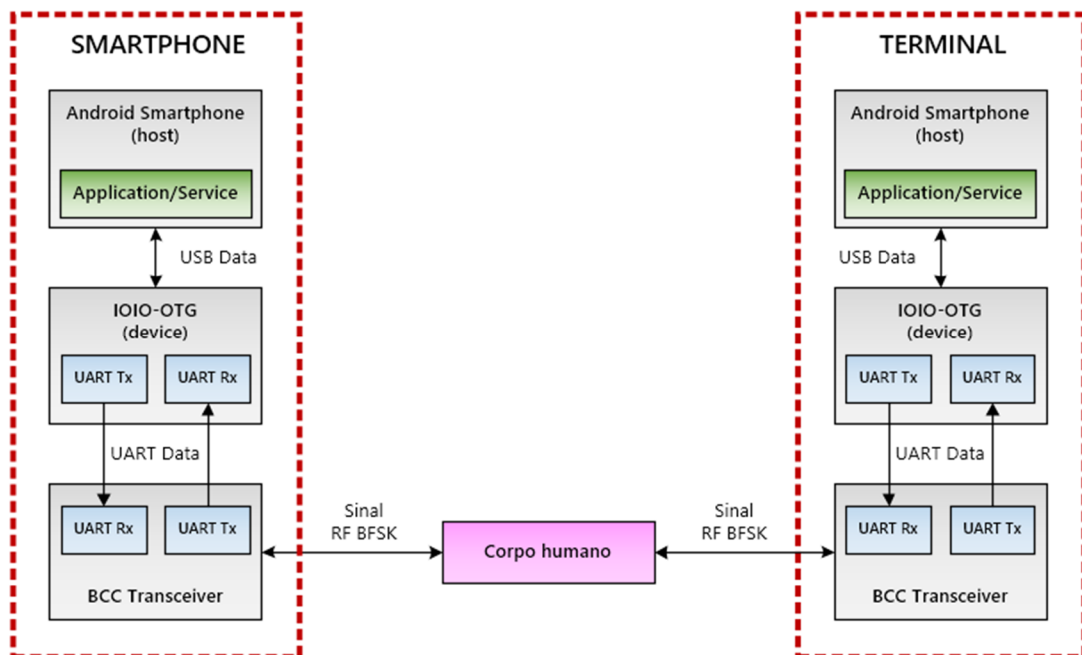


Figura 4.5. Arquitectura detalhada do sistema de pagamento implementado.



Como mostra a figura 4.5, os *smartphones* processam a informação correndo as suas aplicações e trocam a mesma com as placas IOIO-OTG através de uma ligação de dados USB. Por sua vez, as placas IOIO-OTG trocam a informação com os *transceivers* através de uma ligação de dados UART. A informação é depois trocada entre os *transceivers* por acoplamento capacitivo ao corpo humano, segundo o processo descrito no capítulo 3. Naturalmente, durante as várias fases do processo de pagamento ambos os *transceivers* terão que comutar entre os modos emissor e receptor, pois umas vezes terão de enviar a informação e outras vezes terão de recebê-la.

Apesar de o sistema implementado ser ainda um protótipo, o que significa principalmente que este não possui as dimensões desejadas nem tem as suas ligações optimizadas, esta implementação permite ainda assim que o mesmo seja portátil e possa ser anexado ao corpo humano, aproximando-se por isso de um modelo realista. No entanto, uma abordagem ideal seria a que se ilustra na figura 4.6, pelo que neste caso tanto os *transceivers* do sistema BCC como a interface de interligação teriam dimensões reduzidas e estariam integradas no *smartphone* e no terminal de pagamentos. A troca de informação entre o *smartphone* e o *transceiver* não seria efectuada através de uma ligação USB externa nem necessitaria igualmente de uma interface de interligação externa como a que se desenvolveu para o protótipo implementado. Provavelmente, esta troca até poderia ser efectuada de forma directa entre o microprocessador do *smartphone* e o microcontrolador do *transceiver*. Contudo, esta implementação não foi possível de realizar pelos motivos já mencionados, nomeadamente a dificuldade em se obter um terminal de pagamentos que possuísse portas de comunicação I/O e a alta probabilidade de se danificarem os *smartphones* acidentalmente.

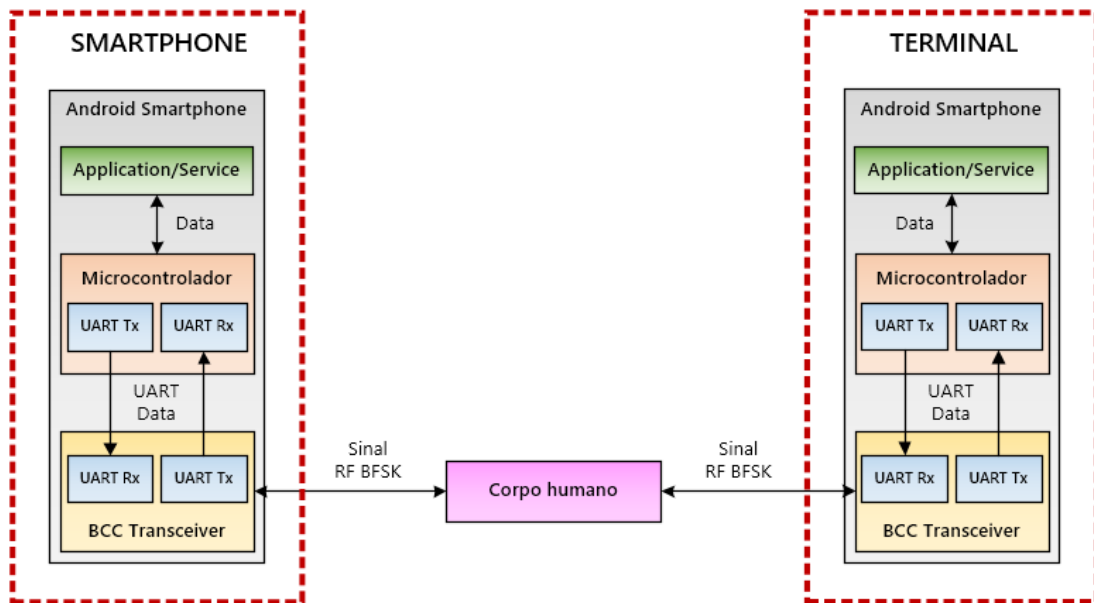


Figura 4.6. Arquitectura ideal para o sistema de pagamento concebido.

Voltando à análise da arquitectura implementada (figura 4.5), durante o processo de um pagamento a comunicação entre o *smartphone* do utilizador (cliente) e o terminal de pagamentos da entidade vendedora é efectuada alternadamente conforme apresentam as figuras 4.7 e 4.8.

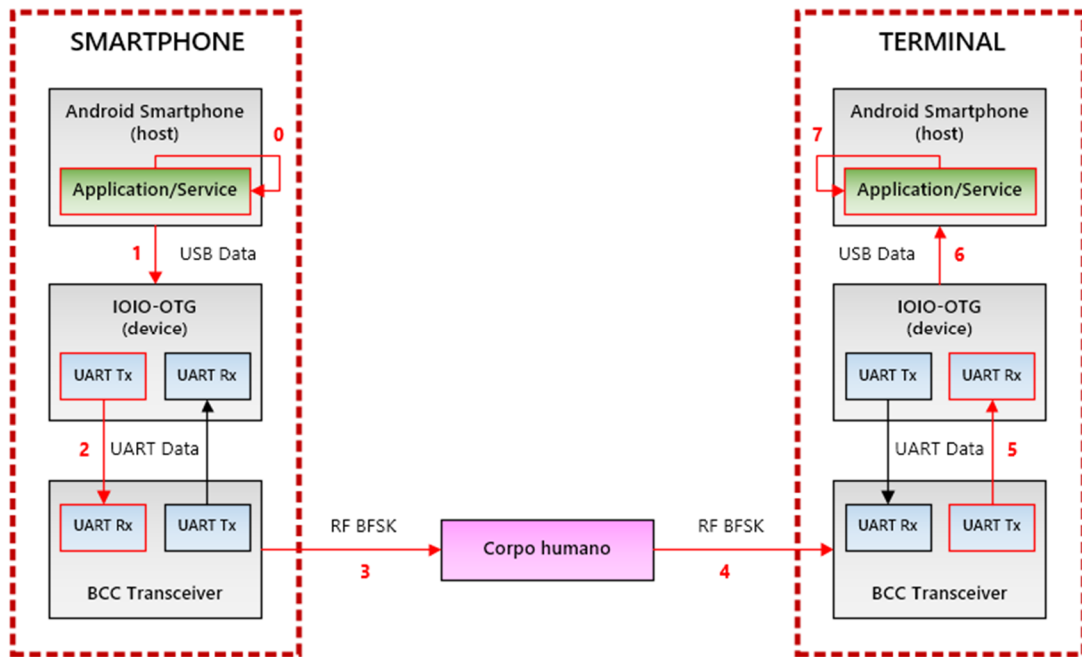


Figura 4.7. Comunicação *smartphone* - terminal.

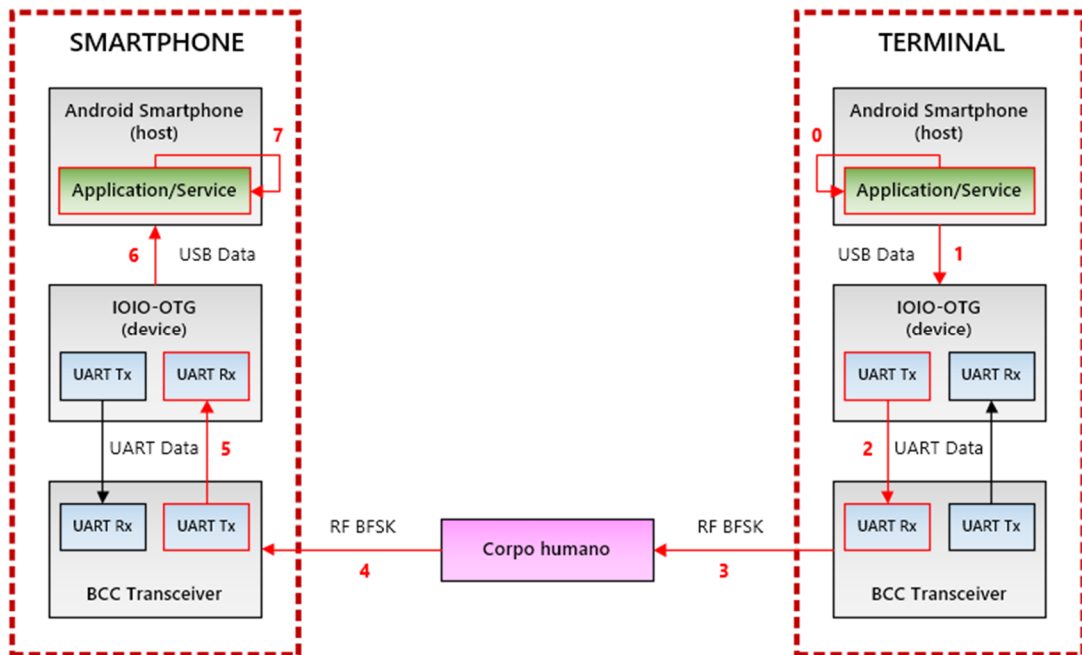


Figura 4.8. Comunicação terminal - *smartphone*.

Pelas figuras 4.7 e 4.8, é possível observar que a comunicação é feita alternadamente entre os dispositivos Android. Portanto, o tipo de comunicação utilizado pode ser visto como um método de *request-response* assíncrono.

Tendo por base a alternância na comunicação, o processo de pagamento do sistema concebido desenrola-se da seguinte forma:

1. A quantia a pagar é inserida no terminal de pagamentos pelo utilizador (cliente) ou pela entidade vendedora.
2. O utilizador ou a entidade vendedora dá início ao processo de pagamento clicando no botão para o efeito que se encontra no terminal.
3. O utilizador mantém-se bastante próximo ou toca no terminal de pagamentos enquanto este último envia um pedido de identificação (*request*) para o *smartphone* do primeiro.
4. O *smartphone* do utilizador recebe o pedido de identificação e responde (*acknowledge*) enviando para o terminal o nome e número de conta do utilizador.
5. O terminal de pagamentos recebe a informação e apresenta uma janela contendo essa informação, a quantia a pagar, o nome e número de conta da entidade vendedora e os botões para confirmação ou cancelamento do pagamento.
6. No caso de a informação estar correcta o utilizador clica no botão de confirmação e de seguida insere o seu PIN na janela apresentada pelo terminal.
7. O utilizador coloca-se novamente bastante próximo ou toca no terminal enquanto este envia o PIN inserido para o *smartphone* do mesmo.
8. O *smartphone* do utilizador recebe o PIN inserido e compara-o com o PIN nele armazenado, pelo que se estiver correcto envia novamente o nome e número de conta do utilizador e um comando específico para o terminal efectuar o pagamento.
9. O terminal de pagamentos recebe a informação, processa-a e efectua o pagamento incrementando o saldo da entidade vendedora e apresentando a factura no ecrã contendo os nomes e números de conta do utilizador e entidade vendedora, a quantia paga e a data do pagamento. De seguida envia uma trama para o *smartphone* do utilizador contendo os dados da factura.
10. O *smartphone* do utilizador recebe os dados e processa-os, decrementa o saldo do utilizador, gera e guarda a factura numa base de dados e apresenta uma notificação que ao ser clicada apresenta uma janela contendo os dados da factura gerada. De seguida, por razões de segurança, o *smartphone* impõe um intervalo de tempo pré-definido que impede que sejam efectuados novos pagamentos enquanto a sua contagem decrescente estiver a decorrer.
11. O processo de pagamento é finalizado, o terminal de pagamentos apresenta a sua janela inicial e o *smartphone* do utilizador volta ao seu estado inicial aguardando por novos pedidos.

## 4.3 Interface Android

### 4.3.1 Interface de interligação

O diagrama de blocos da figura 4.9 representa a interface de interligação implementada para o sistema de pagamento BCC.

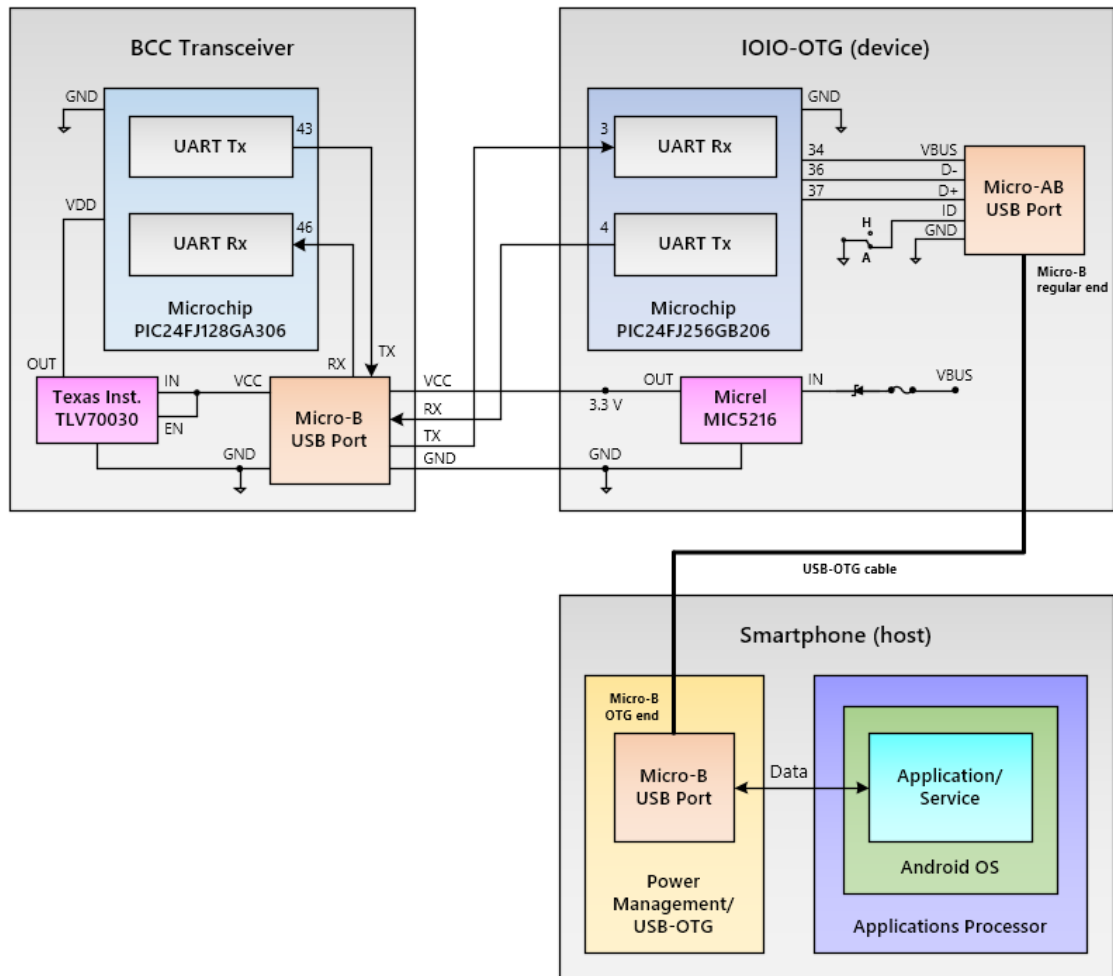


Figura 4.9. Diagrama de blocos da interface de interligação implementada.

Como se pode observar pela figura 4.9, para interligar os *smartphones* às placas IOIO-OTG faz-se uso de uma simples ligação USB, pois as placas já contêm o *firmware* e as bibliotecas necessárias para a comunicação com os *smartphones* Android. Assim, apenas é necessário conectar uma extremidade de um cabo USB-OTG à porta micro-USB do *smartphone* e a outra extremidade à porta micro-USB da placa IOIO-OTG. Obviamente, isto é realizado para cada par *smartphone* - IOIO-OTG.

Existe contudo um factor a ter em conta relativamente às placas IOIO-OTG que se deve ao facto de estas poderem actuar como *host* ou como *device/acessory*. No caso de actuarem como *host* terão de possuir as suas próprias fontes de alimentação (via pino VIN ou conector

JST) e irão alimentar os *smartphones* (*devices*) a que estão conectadas. No caso de actuarem como *device* então podem ser alimentadas pelos próprios *smartphones* (*hosts*) se estes possuírem suporte USB-OTG. Em [57], [58] e [59] podem ser encontrados mais pormenores acerca destes aspectos, pelo que não é objectivo desta dissertação aprofundá-los devido à sua complexidade.

A interface do sistema de pagamento implementado utiliza as placas IOIO-OTG como *device*. Note-se que as portas micro-USB dos *smartphones* e das placas IOIO-OTG são do tipo micro-B e micro-AB, respectivamente, por isso é necessário que a ligação ou o cabo USB utilizado seja do tipo “micro-B-OTG to micro-B”, onde a primeira terminação liga à porta micro-USB do *smartphone* (*host*) e a segunda terminação liga à porta micro-USB da placa IOIO-OTG (*device*). A figura 4.10 apresenta e compara as terminações USB utilizadas para a referida ligação.

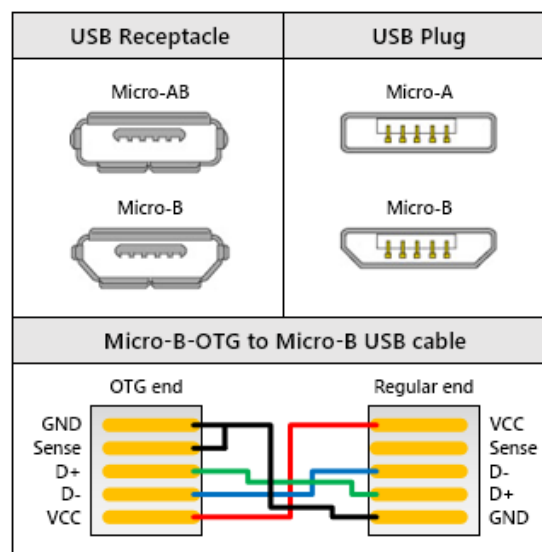


Figura 4.10. Terminações USB utilizadas para a interligação entre os *smartphones* e as placas IOIO-OTG.

Além dos factores anteriormente mencionados, é crucial que o interruptor das placas IOIO-OTG esteja colocado na posição “A” (a posição “H” forçaria o modo *host*) e que os *smartphones* tenham o seu modo de depuração (*debug*) desactivado. Por outro lado, é necessário que a versão do *Application firmware* das placas IOIO-OTG seja a versão 5.0 pois este é o que permite utilizar as bibliotecas Android que por sua vez permitem a utilização das placas como *device*. Consequentemente, por via desta implementação não é utilizada qualquer fonte de alimentação externa para as placas IOIO-OTG pois estas são alimentadas pelos *smartphones*.

A interligação entre as placas IOIO-OTG e os *transceivers* do sistema BCC é feita através de uma ligação de dados UART, pelo que se utilizam dois pinos específicos de cada uma das placas IOIO-OTG para a transmissão e recepção de dados. Como tal, o pino para transmissão de dados UART de cada placa IOIO-OTG é ligado directamente ao pino de recepção de dados UART de cada *transceiver* e vice-versa. O *datasheet* do Microchip PIC24FJ256GB206 [57], microcontrolador incluído na placa IOIO-OTG, indica que este possui 4 módulos UART, pelo que nesta interface utiliza-se apenas um módulo em cada placa IOIO-OTG e a configuração

da trama UART é a mesma que se encontra na figura 3.3, ou seja: 1 start bit (ST), 8 data bits (D1-D7), no parity, 1 stop bit (SP). Pela figura 4.9 observa-se ainda que os *transceivers* BCC são alimentados a partir das placas IOIO-OTG. Deste modo, os *smartphones* alimentam todo o sistema, o que permite aproximá-lo mais do seu modelo ideal e descartar a necessidade de se utilizarem baterias ou outras fontes de alimentação externa. Antes de se proceder à descrição detalhada desta implementação, convém referir primeiro mais alguns aspectos relativos à placa IOIO-OTG.

A partir dos esquemáticos eléctricos das placas IOIO-OTG utilizadas [60] é possível observar que cada uma possui dois reguladores de tensão e um circuito de gestão de carga com potenciómetro, como mostra a figura 4.11. Um dos reguladores de tensão é o Texas Instruments TPS62133, o qual é um conversor *step-down* muito eficiente cujo circuito foi projectado de modo a fornecer à sua saída uma tensão regulada de 5V e uma corrente de 3A a partir de uma gama de tensão de entrada (via pinos VIN ou conector JST) entre 5V e 15V. O outro regulador de tensão é o Micrel MIC5216, o qual é um regulador linear também muito eficiente com capacidade de fornecer altos picos de corrente de saída até 500 mA e cujo circuito foi projectado de modo a colocar à sua saída uma tensão regulada de 3.3V a partir da linha de 5V. O circuito de gestão de carga serve por sua vez para ajustar a quantidade de corrente de carga que é fornecida ao *smartphone* (via linha VBUS) quando a placa IOIO-OTG se encontra configurada como *host*. Permite também que o *smartphone* detecte se está ou não conectado a um *host* e permite gerir o sentido e a intensidade do fluxo de corrente para os dois casos. Além dos circuitos mencionados, é possível observar ainda pela figura 4.11 que a placa IOIO-OTG possui um *header* (JP2) com três pinos que fornecem uma tensão de saída regulada a 3.3V, a qual por sua vez provém da saída do Micrel MIC5216.

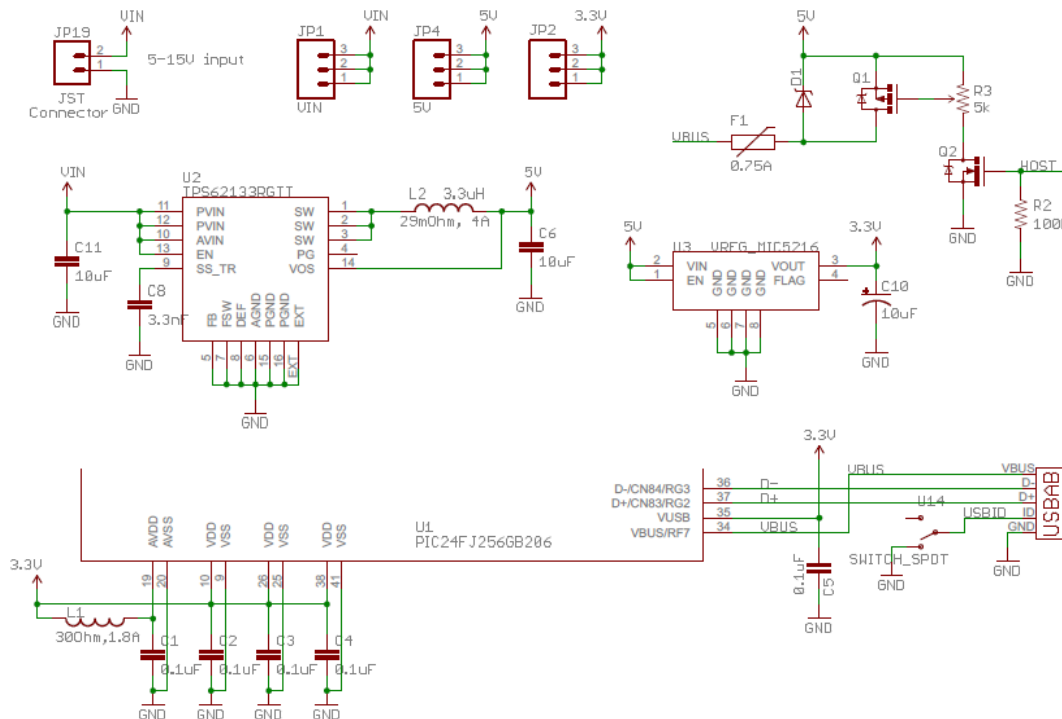


Figura 4.11. Esquemático eléctrico parcial das placas IOIO-OTG utilizadas.

O *transceiver* BCC é alimentado a partir da placa IOIO-OTG ligando directamente um dos pinos do *header* JP2 desta última ao pino de entrada do regulador de tensão Texas Instruments TLV70030 do primeiro.

Esta implementação é possível devido ao facto de o *smartphone* conseguir fornecer potência suficiente para o efeito por via da sua ligação USB com a placa IOIO-OTG. Especificamente, o mesmo disponibiliza na linha VBUS da placa uma tensão e corrente máximas de 5V e 500 mA, respectivamente. Por via da linha VBUS, a potência fornecida é encaminhada para o circuito de gestão de carga e, como neste caso as ligações à direita do díodo deste circuito encontram-se em alta impedância, este por sua vez encaminha-a apenas para o regulador de tensão Micrel MIC5216. Da conversão do regulador resulta uma tensão regulada de 3.3V, a qual além de ser disponibilizada para alimentar *hardware* externo através dos pinos do *header* JP2 da placa é também utilizada para alimentar o microcontrolador desta e os seus respectivos módulos. Como o microcontrolador da placa IOIO-OTG consome pouca corrente (cerca de 30 mA em modo de operação normal), o regulador de tensão Texas Instruments TLV70030 do *transceiver* tem assim disponíveis à sua entrada uma tensão constante de 3.3V e uma corrente máxima em torno dos 400 mA, algo mais que suficiente para alimentar os *transceivers* pois o mesmo foi projectado para fornecer à sua saída uma tensão regulada de 3V e uma corrente máxima de 200 mA.

Não é demais referir que caso não fosse possível seguir por esta via, haveria outra solução possível, a qual consistiria em utilizar uma fonte de alimentação externa (bateria por exemplo) entre 5V a 15V que alimentaria a placa IOIO-OTG através do seu conector JST ou dos seus pinos VIN. Assim, haveriam três ramificações possíveis para a sua implementação.

A primeira seria alimentar *transceiver* a partir da placa IOIO-OTG por via dos pinos do *header* JP2 segundo o processo já descrito anteriormente.

A segunda seria alimentar o *transceiver* também através da placa IOIO-OTG mas desta vez a partir dos pinos do *header* JP4, os quais disponibilizam uma tensão de saída regulada a 5V proveniente da conversão do Texas Instruments TPS62133. Tal como para os casos anteriores, esta saída seria ligada ao pino de entrada do regulador de tensão Texas Instruments TLV70030 do *transceiver*.

A terceira e última possibilidade passaria por alimentar individualmente tanto a placa IOIO-OTG como o *transceiver*. Portanto, ambos teriam a sua própria fonte de alimentação externa, pelo que no caso do *transceiver* essa fonte seria uma pilha do tipo botão.

Menciona-se ainda que para todos estes casos a placa IOIO-OTG deveria ser configurada como *host* colocando-se o seu interruptor na posição “H” e utilizando uma ligação USB do tipo “micro-B to micro-A”, onde a primeira terminação liga à porta micro-USB do *smartphone* (*device*) e a segunda terminação liga à porta micro-USB da placa IOIO-OTG (*host*).

Apesar de mais simples e imediatas, estas soluções afastariam o sistema de pagamento desenvolvido do seu modelo ideal pois iriam sacrificar o seu tamanho e a sua portabilidade, além do acréscimo de componentes que trariam, nomeadamente fontes de alimentação externa.

Em nota final, o *software* base que é necessário desenvolver em Android para a interacção entre as placas IOIO-OTG e os *smartphones* será explicado na secção 4.4.4 (IOIO-OTG).

### 4.3.2 Terminal de pagamentos

O terminal de pagamentos implementado consiste num *smartphone* equipado com Android e com a respectiva aplicação que o emula. A aplicação desenvolvida permite efectuar as

operações necessárias à realização de um pagamento seguro através do corpo humano desde que a comunicação com o *smartphone* (carteira digital) do utilizador seja bem-sucedida. A aplicação designa-se por BCC Terminal e o seu modo de funcionamento é o que se apresenta de forma simplificada no fluxograma da figura 4.12.

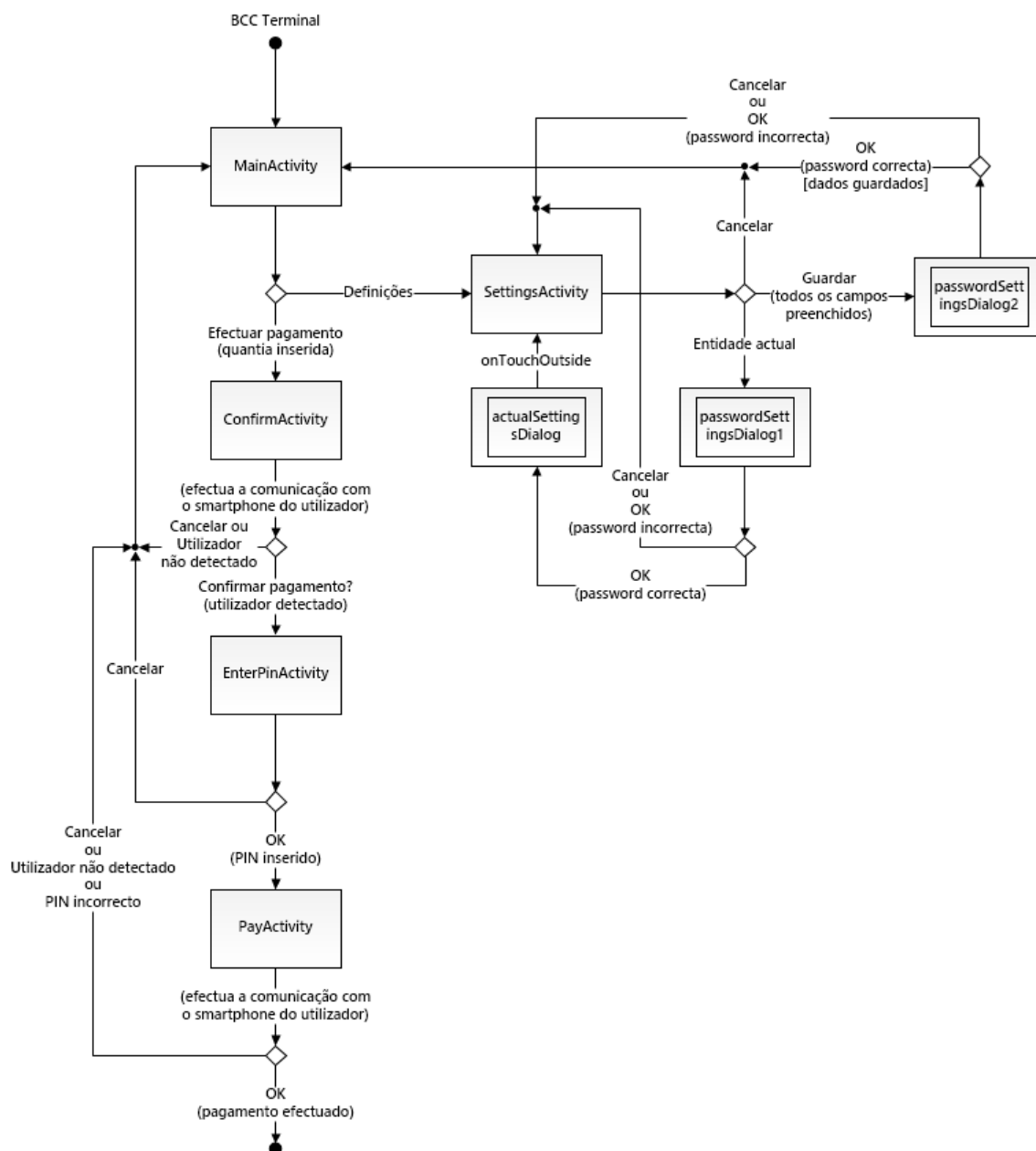


Figura 4.12. Fluxograma do funcionamento da aplicação BCC Terminal.

Como explica o fluxograma da figura 4.12, ao se abrir a aplicação BCC Terminal é apresentada em primeiro lugar uma janela (MainActivity) que emula um terminal de pagamentos no qual a entidade vendedora pode inserir a quantia a pagar pelo utilizador (cliente). Ao se inserir uma quantia válida e depois de o utilizador clicar no botão “Efectuar pagamento”, sur-



ge uma nova janela (ConfirmActivity) na qual o terminal detecta e identifica esse mesmo utilizador via comunicação através do corpo humano. Se o utilizador foi correctamente detectado e identificado, surge no ecrã a informação relevante ao pagamento (montante inserido e também os nomes e números de conta da entidade e do utilizador) e os botões que permitem confirmar ou cancelar o pagamento, caso contrário, o pagamento é automaticamente cancelado. Ao ser clicado o botão “Confirmar pagamento?” é apresentada uma nova janela (EnterPinActivity) que solicita a inserção do PIN do utilizador identificado. Depois de inserido o PIN e de ser clicado o botão “OK”, surge outra janela (PayActivity) na qual mais uma vez o terminal detecta e identifica o utilizador e verifica se o PIN inserido está correcto, via comunicação através do corpo humano. Caso contrário, o pagamento é também aqui cancelado automaticamente. No caso de o PIN estar correcto, o terminal credita o saldo da conta da entidade vendedora com o montante inserido e regista a data do pagamento, apresentando logo de seguida no ecrã a factura com as informações relevantes (montante pago, nomes e números de conta da entidade e do utilizador e a data do pagamento). É importante destacar que durante esta última fase do pagamento, o terminal envia para o *smartphone* (carteira digital) do utilizador as informações necessárias à conclusão do mesmo, entre elas, os dados da entidade vendedora (nome e número de conta), o montante a debitar na conta do utilizador e a data registada. Se ocorrer algum erro na comunicação o pagamento é automaticamente cancelado, tal como nas fases anteriores.

Por fim, falta referir que na primeira janela (MainActivity) existe também um botão de preferências designado por “Definições” que permite configurar os dados da entidade vendedora no terminal de pagamentos. Ao clicar esse botão surge a janela das preferências (SettingsActivity) na qual se pode inserir o nome e o número de conta da entidade bem como a *password* para proteger os dados inseridos. Dado que a aplicação desenvolvida é ainda um protótipo, nesta janela existe também um campo que permite inserir o saldo da conta da entidade, apenas para efeitos de teste. Para visualizar os dados da entidade actual basta clicar no botão “Entidade actual” e inserir a *password* correcta, pelo que surgirá uma subjanela com esses dados.

Nesta secção apenas se pretende efectuar uma descrição simplificada da aplicação BCC Terminal, pelo que a explicação mais detalhada do *software* desenvolvido para a mesma encontra-se na secção 4.4.5 (BCC Terminal).

O diagrama da figura 4.13 procura dar uma visão geral da aplicação e ilustrar o seu funcionamento de forma mais intuitiva.



Figura 4.13. Diagrama com as janelas da aplicação BCC Terminal.

### 4.3.3 Carteira digital

A carteira digital do utilizador implementada consiste num *smartphone* equipado com Android e com a respectiva aplicação para o efeito. A aplicação desenvolvida permite efectuar as operações necessárias à realização de um pagamento seguro através do corpo humano desde que a comunicação com o *smartphone* (terminal de pagamentos emulado) da entidade vendedora seja bem-sucedida. A aplicação designa-se por BCC Wallet e o seu modo de funcionamento é o que se apresenta de forma simplificada no fluxograma da figura 4.14.

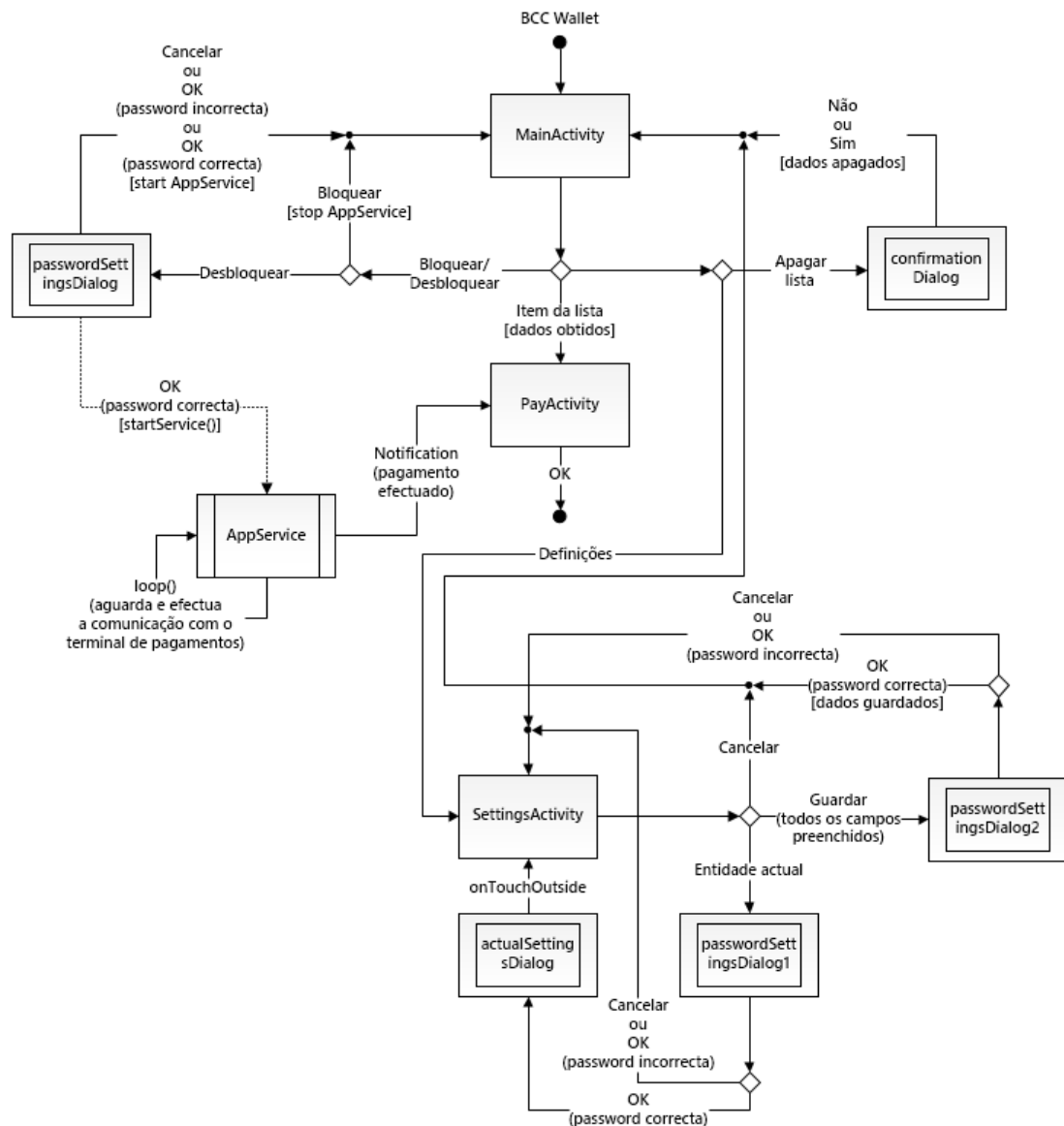


Figura 4.14. Fluxograma do funcionamento da aplicação BCC Wallet.

Como explica o fluxograma da figura 4.14, ao se abrir a aplicação BCC Wallet é apresentada em primeiro lugar uma janela (MainActivity) que apresenta as informações relativas ao utilizador, como o seu saldo actual, o seu nome e uma lista com todos os pagamentos efectua-

dos. Além disso, apresenta um botão (designado por “Bloquear/Desbloquear”) que permite bloquear ou desbloquear o serviço para efectuar pagamentos, protegido pela introdução do PIN do utilizador, aumentando desta forma a segurança da aplicação. Existem ainda dois botões na barra superior da janela, sendo eles, um para apagar a lista dos pagamentos e outro para abrir a janela das preferências. Ao se clicar no primeiro, designado por “Apagar lista”, surge uma subjanela que pergunta ao utilizador se este pretende apagar a lista, pelo que a mesma será apagada se o utilizador clicar no botão de confirmação “Sim”. Ao se clicar no segundo, designado por “Definições”, surge a janela das preferências (SettingsActivity), a qual é semelhante à que foi desenvolvida para a aplicação BCC Terminal e permite realizar as mesmas operações que foram explicadas na secção anterior. Como é lógico, nesta aplicação a informação envolvida nessas operações será desta vez relativa ao utilizador.

Ao ser clicada uma das linhas da lista dos pagamentos efectuados da janela principal, surge uma outra janela (PayActivity) que contém os dados relativos a esse pagamento, isto é, que apresenta a factura desse pagamento. Deste modo, a aplicação permite ver as facturas de todos os pagamentos que tenham sido efectuados.

No caso de o utilizador ter a sua carteira digital desbloqueada, ou seja, no caso de a aplicação se encontrar no estado em que permite efectuar pagamentos, então a mesma correrá um serviço em *background* no sistema operativo que efectuará todas as operações necessárias ao processamento de um pagamento. O serviço, enquanto activo, aguardará pela comunicação de um terminal de pagamentos (neste caso emulado por um *smartphone*) e fará a comunicação com o mesmo até o pagamento estar concluído. Durante a comunicação, o serviço processa as informações recebidas que são relativas à entidade vendedora e ao pagamento em questão e envia para o terminal os dados relativos ao utilizador (cliente). Caso não ocorram erros, no final o serviço decrementa o saldo do utilizador e envia uma notificação para a barra de notificações do sistema operativo a indicar que o pagamento foi concluído, pelo que ao ser clicada essa notificação surgirá a janela (PayActivity) com a factura desse pagamento.

Tal como para a secção anterior, nesta secção apenas se pretende efectuar uma descrição simplificada da aplicação BCC Wallet, pelo que a explicação mais detalhada do *software* desenvolvido para a mesma encontra-se na secção 4.4.6 (BCC Wallet).

Os diagramas das figuras 4.15 e 4.16 procuram dar uma visão geral da aplicação e ilustrar o seu funcionamento de uma forma mais intuitiva.

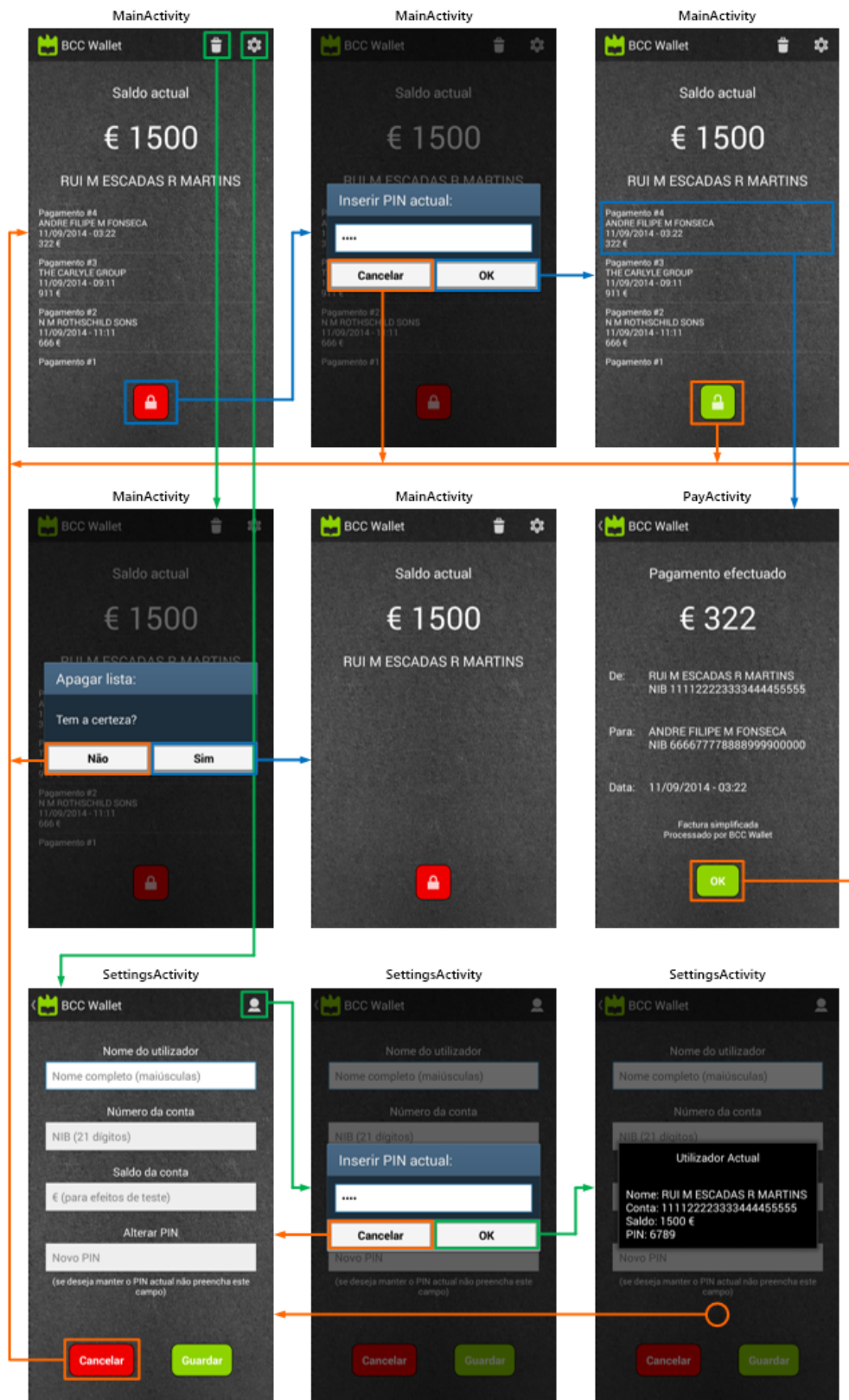


Figura 4.15. Diagrama com as janelas da aplicação BCC Wallet.

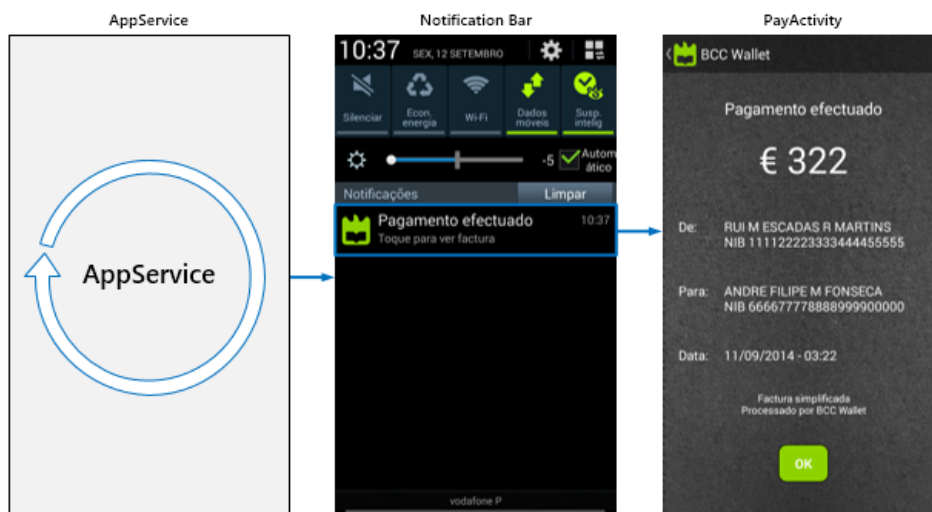


Figura 4.16. Diagrama ilustrativo de como é processado um pagamento pela aplicação BCC Wallet (ponto de vista do utilizador).

## 4.4 Software e estruturação dos dados

### 4.4.1 Entidade vendedora

Antes de se desenvolverem as aplicações Android do sistema de pagamento, definiram-se as estruturas das entidades intervenientes no processo de pagamento, sendo elas, a entidade vendedora e o utilizador (cliente). Além disso, também se definiu a estrutura de um pagamento. A entidade vendedora é estruturada da forma que se apresenta na figura 4.17.

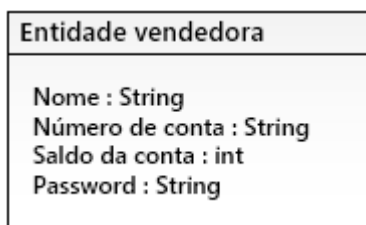


Figura 4.17. Entidade vendedora.

Como se pode observar pela figura 4.17, a entidade vendedora tem a si associada um nome, um número de conta, o saldo dessa mesma conta e uma *password*. Tanto o nome como o número de conta servem para identificá-la e torná-la única. Por sua vez, o saldo da conta define o montante existente na conta da entidade e a *password* serve para proteger os seus dados no terminal de pagamentos. O nome, o número de conta e a *password* terão a si associadas variáveis do tipo *String* e o saldo da conta uma variável do tipo *int*.

#### 4.4.2 Utilizador (cliente)

De modo semelhante à secção anterior, nesta secção define-se a estrutura do utilizador (cliente). Assim, o utilizador (cliente) é estruturado da forma que se apresenta na figura 4.18.

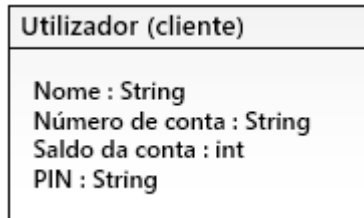


Figura 4.18. Utilizador (cliente).

Como se pode observar pela figura 4.18, o utilizador (cliente) tem a si associado um nome, um número de conta, o saldo dessa mesma conta e um PIN. Tanto o nome como o número de conta servem para identificá-lo e torná-lo único. Por sua vez, o saldo da conta define o montante existente na conta do utilizador e o PIN serve para proteger os seus dados na sua carteira digital e para fornecer um acréscimo de segurança durante o acto de um pagamento. O nome, o número de conta e o PIN terão a si associadas variáveis do tipo *String* e o saldo da conta uma variável do tipo *int*.

#### 4.4.3 Pagamento

A figura 4.19 apresenta a estrutura considerada para cada pagamento, os quais serão armazenados na base de dados da carteira digital do utilizador.

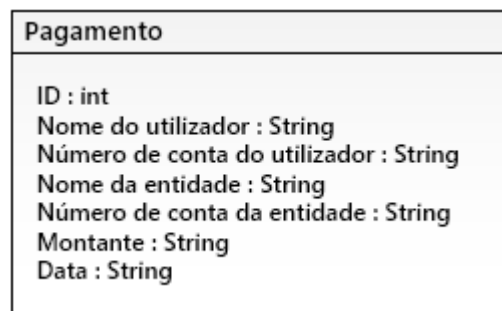


Figura 4.19. Pagamento.

Pela figura 4.19 é possível observar que um pagamento tem a si associado um número (*id*) e os nomes e números de conta do utilizador (cliente) e da entidade vendedora, o montante pago e a data do pagamento. Por sua vez, o *id* tem a si associado uma variável do tipo *int* e os restantes objectos variáveis do tipo *String*.

#### 4.4.4 IOIO-OTG

Definida a estruturação dos dados, é necessário ter algumas considerações prévias no que toca ao desenvolvimento de uma *Activity* ou *Service* em Android que implemente uma comunicação entre um *smartphone* e uma placa IOIO-OTG. Nesta secção é apresentado e explicado o código base (em linguagem de programação Java) que é necessário desenvolver para tornar possível essa comunicação. Em pequena nota, convém apenas referir que, a partir da presente secção, qualquer código que for irrelevante para cada análise será representado por “(...)”.

Em primeiro lugar, terão de ser importadas para o projecto as bibliotecas desenvolvidas para Android que implementam a plataforma base para a interacção com a placa IOIO-OTG, isto é, que contêm as classes e métodos que permitem interagir com esta última. Essas bibliotecas são as seguintes:

- IOIOLibAccessory
- IOIOLibAndroid
- IOIOLibAndroidDevice

Assim, a declaração de uma *Activity* na qual se pretenda a interacção com a placa IOIO-OTG deverá ser realizada da forma que se ilustra na tabela 4.1.

```
public class MainActivity extends IOIOActivity {}
```

Tabela 4.1. Criação de uma subclasse da IOIOActivity.

Pelo código da tabela 4.1 observa-se que deverá ser criada uma subclasse da classe IOIOActivity que irá implementar a plataforma para a interacção com a placa IOIO-OTG. Deste modo, o programador poderá focar-se apenas no código específico para a aplicação a desenvolver, pois poderá usar directamente os métodos já existentes para a interacção com a placa IOIO-OTG, pelo que não terá de os desenvolver.

Contudo, um dos requisitos em Android é o facto de a *UiThread* da aplicação não poder ser bloqueada, pois se esta o for durante 5 segundos a mesma será terminada pelo sistema operativo sob a forma de uma excepção fatal. Como tal, o processamento de toda a actividade relativa à placa IOIO-OTG terá de ser efectuado numa subclasse da classe BaseIOIOLooper que por sua vez será declarada dentro da *Activity* em questão. A tabela 4.2 ilustra como deverá ser criada esta subclasse.

```
public class MainActivity extends IOIOActivity
{
    (...)

    class Looper extends BaseIOIOLooper
    {
        (...)

        @Override
        protected void setup() throws ConnectionLostException
        {
            (...)
        }
    }
}
```



```

@Override
protected void loop() throws ConnectionLostException
{
    (...)
    try {
        Thread.sleep(100);
    }
    catch (InterruptedException e) {}
}
} (...)
}

```

Tabela 4.2. Criação de uma subclasse da BaseIOIOLooper.

Como é possível observar no código da tabela 4.2, existem dois métodos nesta classe que terão de ser implementados. O primeiro é o método *setup()*, o qual é invocado sempre que o *smartphone* detecta a placa IOIO-OTG e que tem como objectivo efectuar todas as configurações da placa necessárias à aplicação. O segundo é o método *loop()*, o qual consiste num ciclo infinito que é invocado logo após o método *setup()* e que será executado indefinidamente. Repare-se que dentro deste método terá de ser sempre invocado o método *sleep()*, pois este é o que permitirá suspender periodicamente esta *thread* de modo a que as outras *threads* tenham possibilidade de serem executadas. O método *sleep()* encontra-se dentro de uma construção *try/catch*, pelo que qualquer excepção que ocorra a partir do mesmo será ignorada.

Por fim, é necessário incluir na *Activity* o código que se apresenta na figura 4.3, o qual irá criar a *thread* que processa a actividade relativa à placa IOIO-OTG.

```

public class MainActivity extends IOIOActivity
{
    (...)

    @Override
    protected IOIOLooper createIOIOLooper()
    {
        return new Looper();
    }
}

```

Tabela 4.3. Criação da *thread* para o processamento relativo à placa IOIO-OTG.

Como nota final, refere-se que para o caso de um *Service* este procedimento seria equivalente, pelo que apenas teria de se substituir “Activity” por “Service” no código apresentado nas tabelas 4.1, 4.2 e 4.3.

#### 4.4.5 BCC Terminal

Dada a complexidade do *software* desenvolvido para este sistema de pagamento, nesta secção apenas se destacam os seus aspectos mais relevantes no que toca à aplicação BCC Terminal. A aplicação foi desenvolvida utilizando o programa Eclipse ADT (versão 23.0.2), o qual consiste numa versão do Eclipse IDE com o *package* ADT integrado e com todos os componentes necessários para o desenvolvimento em Android. As linguagens de programação utilizadas foram Java e XML. Para testar a aplicação foi utilizado um *smartphone* Samsung Galaxy S3 equipado com Android 4.3 (Jelly Bean).

Como foi possível observar na secção 4.3.2, a aplicação BCC Terminal é composta por cinco *Activities*, sendo elas:

- MainActivity
- SettingsActivity
- ConfirmActivity
- EnterPinActivity
- PayActivity

A MainActivity é a primeira *Activity* que é invocada logo após se abrir a aplicação e permite à entidade vendedora ou ao utilizador inserir a quantia a pagar e dar início ao processo de pagamento através dos eventos associados aos *Buttons* implementados. Ao ser clicado o *Button* que dá início ao processo de pagamento, é invocada logo de seguida a ConfirmActivity passando-se um *Intent* contendo uma variável do tipo *int* associada à quantia inserida. A MainActivity permite ainda aceder à SettingsActivity por via do *Button* implementado para o efeito na sua *Action Bar*.

A SettingsActivity permite configurar os dados da entidade vendedora e visualizar os dados actuais. A visualização dos dados actuais é apresentada via *Dialog*, o qual é invocado a partir do *Button* implementado na sua *Action Bar*. De modo a proporcionar maior segurança, essa visualização é protegida por um *AlertDialog* que solicita a inserção da *password* actual da entidade vendedora, a qual se encontra associada a uma variável do tipo *String* armazenada numa zona reservada do directório de dados da aplicação através do uso da classe *SharedPreferences*. A configuração ou inserção de novos dados é realizada preenchendo todos os campos das *EditTexts* implementadas e clicando no *Button* para esse efeito. Antes de serem guardados os novos dados também irá surgir um *AlertDialog* que solicita a inserção da *password* actual da entidade vendedora e, no caso de ser inserida correctamente, os dados serão armazenados recorrendo a métodos associados à classe *SharedPreferences*. O nome da entidade, o número de conta e o saldo da conta terão a si associados variáveis do tipo *String*, *String* e *int*, respectivamente.

A ConfirmActivity, invocada a partir da MainActivity, tem como função identificar e detectar o utilizador e servir de janela de confirmação para o processo de pagamento. Para aumentar a segurança foi implementado um temporizador que cancela o processo de pagamento automaticamente após ser atingido o tempo limite pré-definido (15 segundos). Este foi implementado conforme se apresenta na tabela 4.4.

```
public class ConfirmActivity extends IOIOActivity
{
    (...)
    public CountdownTimer countdownTimer;
    public int countdown_time = 15;

    @Override
    protected void onCreate(Bundle savedInstanceState)
    {
        (...)

        countdownTimer = new CountdownTimer(countdown_time*1000, 1000)
        {
```

```

        public void onTick(long millisUntilFinished) {}
        public void onFinish()
        {
            Toast toast = Toast.makeText(ConfirmActivity.this,
            "Utilizador não detectado", Toast.LENGTH_LONG);
            toast.show(); finish();
        }
    }.start();
} (...)
}

```

Tabela 4.4. Temporizador para cancelamento do processo de pagamento.

A detecção do utilizador é feita através do *loop()* implementado para a comunicação entre os *smartphone* e a placa IOIO-OTG e, caso a comunicação seja bem-sucedida, a identificação do utilizador é depois efectuada recorrendo a uma *AsyncTask*, o que permite diminuir o esforço computacional. Note-se que antes de ser feita a detecção e identificação do utilizador, a par da inicialização de todas as variáveis são também efectuadas todas as configurações relativas à placa IOIO-OTG e à trama de dados UART, como exemplifica o código apresentado na tabela 4.5.

```

public class ConfirmActivity extends IOIOActivity
{
    (...)

    class Looper extends BaseIOIOLooper
    {
        private static final int RX_PIN = 3;
        private static final int TX_PIN = 4;
        private static final int BAUD_RATE = 9600;
        private Uart uart;
        private InputStream stream_in = null;
        private OutputStream stream_out = null;

        (...)

        @Override
        protected void setup() throws ConnectionLostException
        {
            uart = ioio_.openUart(RX_PIN, TX_PIN, BAUD_RATE,
            Uart.Parity.NONE, Uart.StopBits.ONE);
            stream_in = uart.getInputStream();
            stream_out = uart.getOutputStream();
        } (...)
    }
}

```

Tabela 4.5. Configuração da trama de dados UART para a placa IOIO-OTG.

O processo de detecção do utilizador encontra-se no código da tabela 4.6.

```

public class ConfirmActivity extends IOIOActivity
{
    public String received_string = "";

    (...)

    class Looper extends BaseIOIOLooper
    {
        private char received_char;
    }
}

```

```

private boolean done = false, start_string = false, end_string = false;

(...)

@Override
public void loop() throws ConnectionLostException
{
    try {
        Thread.sleep(100);
    }
    catch (InterruptedException e) {}

    if (done == false)
    {
        try {
            if (start_string == false)
                stream_out.write('*');
        }
        catch (IOException e) {}

        try {
            received_char = (char) stream_in.read();
        }
        catch (IOException e) {}

        if (received_char == '[')
            start_string = true;
        if (start_string == true)
            received_string = received_string + received_char;
        if (received_char == ']')
            end_string = true;
        if (start_string == true && end_string == true)
        {
            done = true;
            uart.close();
            runOnUiThread(new Runnable() {
                public void run()
                {
                    countdownTimer.cancel();
                    new searchUser().execute();
                }
            });
        }
    }
}
} (...)
} (...)
}

```

Tabela 4.6. Detecção do utilizador (ConfirmActivity).

Como se pode observar na tabela 4.6, o terminal envia primeiro um *request* para o *smartphone* do utilizador através de um carácter específico via UART. Depois, o terminal recebe novamente via UART uma trama contendo os dados solicitados (nome e número de conta do utilizador) no caso de o *acknowledge* por parte do utilizador ter sido bem-sucedido. À medida que os caracteres da trama vão sendo recebidos, os mesmos vão sendo armazenados sequencialmente numa variável do tipo *String*. A delimitação da informação solicitada e a separação dos campos pretendidos nessa variável é efectuada utilizando caracteres especiais, como apresenta a figura 4.20.

[	Nome do utilizador	#	Número de conta do utilizador	]
---	--------------------	---	-------------------------------	---

Figura 4.20. Trama enviada pelo *smartphone* do utilizador (identificação).

Após toda a informação ser recebida, a ligação de dados UART é terminada e é invocada a *AsyncTask* implementada para a identificação do utilizador. O código da tabela 4.7 mostra como é efectuado o processo de identificação do utilizador.

```
public class ConfirmActivity extends IOIOActivity
{
    (...)

    private class searchUser extends AsyncTask<Void, Void, Boolean>
    {
        protected Boolean doInBackground(Void... params)
        {
            if (received_string.contains("[") && received_string.contains("#")
            && received_string.contains("]"))
            {
                received_string = received_string.replace("[", "");
                received_string = received_string.replace("]", "");
                String[] parts = received_string.split("#");
                receivedUserName = parts[0];
                receivedUserAccount = parts[1];
                receivedUserString = true;
            }
            else
            {
                receivedUserString = false;
            }
            return receivedUserString;
        }

        protected void onPostExecute(Boolean receivedUserString)
        {
            if (receivedUserString == true)
            {
                loop.setVisibility(View.INVISIBLE);
                processing.setVisibility(View.INVISIBLE);
                final SharedPreferences pref =
                getSharedPreferences("BCCTerminalPrefs", MODE_PRIVATE);
                storedEntityName = pref.getString("storedEntityName", "-");
                storedEntityAccount = pref.getString("storedEntityAccount",
                "-");
                receive_amount.setVisibility(View.VISIBLE);
                received_amount.setVisibility(View.VISIBLE);
                received_amount.setText("€ " + String.valueOf(amount));
                from.setVisibility(View.VISIBLE);
                to.setVisibility(View.VISIBLE);
                inserted_from.setVisibility(View.VISIBLE);
                inserted_from.setText(receivedUserName + "\nNIB " +
                receivedUserAccount);
                inserted_to.setVisibility(View.VISIBLE);
                inserted_to.setText(storedEntityName + "\nNIB " +
                storedEntityAccount);
                confirm_button.setVisibility(View.VISIBLE);
            }
            else
            {
                Toast toast = Toast.makeText(ConfirmActivity.this,
                "Utilizador não detectado", Toast.LENGTH_LONG);
                toast.show(); finish();
            }
        }
    }

    (...)
}
```

Tabela 4.7. Identificação do utilizador (ConfirmActivity).

Como é possível observar, ao ser executada a *AsyncTask* os campos da informação armazenada relativa ao utilizador (nome e número de conta do utilizador) são separados e armazenados individualmente em variáveis do tipo *String*. Essa informação é depois apresentada no ecrã a par da informação relativa à entidade vendedora.

Para confirmar o pagamento o utilizador apenas terá de clicar o *Button* implementado para o efeito, pelo que a *Activity* *EnterPinActivity* será invocada passando-se um *Intent* contendo as variáveis associadas à quantia inserida, ao nome do utilizador e ao número de conta deste.

A *EnterPinActivity* permite reforçar a segurança do sistema de pagamento ao solicitar a inserção do PIN do utilizador. Este último pode inserir o seu PIN e concluir a sua identificação através dos *Buttons* implementados para o efeito. O PIN inserido será armazenado numa variável temporária do tipo *String* e cada dígito será ocultado no ecrã do *smartphone* por um carácter especial de modo a que os mesmos não sejam visíveis a terceiros. Ao concluir-se esta fase, será invocada a *PayActivity* passando-se um *Intent* contendo as variáveis associadas à quantia inserida, ao PIN inserido, ao nome do utilizador e ao número de conta deste.

A *PayActivity* é a *Activity* que conclui o processo de pagamento. Para aumentar a segurança foi implementado outro temporizador que cancela o processo de pagamento automaticamente após ser atingido o tempo limite pré-definido (15 segundos) e é efectuada novamente a detecção e identificação do utilizador mas desta vez a informação recebida é comparada com os dados recebidos na *ConfirmActivity* de modo a impossibilitar que a confirmação e a conclusão do processo de pagamento seja realizada por dois utilizadores diferentes. A tabela 4.8, mostra como é efectuado o processo de detecção do utilizador.

```
public class PayActivity extends IOIOActivity
{
    (...)

    class Looper extends BaseIOIOLooper
    {
        (...)

        @Override
        public void loop() throws ConnectionLostException
        {
            (...)

            if (done == false)
            {
                try {
                    if (start_string == false)
                        stream_out.write("@ " +
                            pin_string).getBytes();
                } catch (IOException e) {}

                try {
                    received_char = (char) stream_in.read();
                } catch (IOException e) {}

                if (received_char == '%')
                {
                    runOnUiThread(new Runnable() {
                        public void run()
                        {
                            Toast toast2 =
                                Toast.makeText(PayActivity.this, "PIN
                                incorrecto", Toast.LENGTH_LONG);
                            toast2.show();
                        }
                    });
                }
            }
        }
    }
}
```

```

        finish();
    }
    });
}
if (received_char == '[')
    start_string = true;
if (start_string == true)
    received_string = received_string + received_char;
if (received_char == ']')
    end_string = true;
if (start_string == true && end_string == true)
{
    done = true;
    runOnUiThread(new Runnable() {
        public void run()
        {
            new idUser().execute();
        }
    });
}
}
} (...)
} (...)
}

```

Tabela 4.8. Detecção do utilizador (PayActivity).

Como se observa na tabela 4.8, o processo de detecção do utilizador é feito de forma semelhante à ConfirmActivity, pelo que se implementou novamente o *loop()* que permite a comunicação entre o terminal de pagamentos e o *smartphone* do utilizador. Antes de efectuar a comunicação, a par da inicialização de todas as variáveis também é configurada a ligação de dados UART de forma semelhante à que se apresenta na tabela 4.5. De seguida, o terminal envia um *request* para o *smartphone* do utilizador através de um carácter específico via UART seguido pelos caracteres do PIN inserido pelo utilizador, como ilustra a figura 4.21.

@	PIN inserido
---	--------------

Figura 4.21. Trama enviada pelo terminal de pagamentos (identificação).

Depois, o terminal recebe novamente via UART uma trama específica contendo os dados solicitados no caso de o *acknowledge* por parte do utilizador ter sido bem-sucedido. Note-se que no caso de o PIN estar incorrecto, o terminal recebe um carácter específico que dará início ao cancelamento do pagamento. À medida que os caracteres da trama vão sendo recebidos, os mesmos vão sendo armazenados sequencialmente numa variável do tipo *String* onde a delimitação da informação solicitada e a separação dos campos pretendidos nessa variável é feita de forma semelhante à ConfirmActivity, como apresenta a figura 4.22. Repare-se que desta vez existe mais um campo na trama que contém um comando específico para indicar que o pagamento deverá ser efectuado.

[	Nome do utilizador	#	Número de conta do utilizador	#	PAY	]
---	--------------------	---	-------------------------------	---	-----	---

Figura 4.22. Trama enviada pelo *smartphone* do utilizador (confirmação).

Após esta informação ser recebida é invocada a *AsyncTask* implementada para a identificação do utilizador, a qual se encontra na tabela 4.9.

```
public class PayActivity extends IOIOActivity
{
    (...)

    @Override
    protected void onCreate(Bundle savedInstanceState)
    {
        (...)
        Intent intent = getIntent();
        receivedUserName = intent.getStringExtra("receivedUserName");
        receivedUserAccount = intent.getStringExtra("receivedUserAccount");
        pin_string = intent.getStringExtra("pin_string");
        amount = intent.getIntExtra("amount", amount);
        (...)
    } (...)

    private class idUser extends AsyncTask<Void, Void, Boolean>
    {
        protected Boolean doInBackground(Void... params)
        {
            if (received_string.contains("[") && received_string.contains("#")
            && received_string.contains("]"))
            {
                received_string = received_string.replace("[", "");
                received_string = received_string.replace("]", "");
                String[] parts = received_string.split("#");
                receivedUserNameAgain = parts[0];
                receivedUserAccountAgain = parts[1];
                String pay = parts[2];
                if (receivedUserNameAgain.equals(receivedUserName) &&
                receivedUserAccountAgain.equals(receivedUserAccount) &&
                pay.equals("PAY"))
                    receivedUserString = true;
                else receivedUserString = false;
            }
            else
            {
                receivedUserString = false;
            }
            return receivedUserString;
        }

        protected void onPostExecute(Boolean receivedUserString)
        {
            if (receivedUserString == true) {}
            else
            {
                Toast toast2 = Toast.makeText(PayActivity.this, "Operação
                anulada", Toast.LENGTH_LONG);
                toast2.show();
                finish();
            }
        }
    } (...)
}
```

Tabela 4.9. Identificação do utilizador (PayActivity).

Tal como na ConfirmActivity, ao ser executada a *AsyncTask* os campos da informação armazenada são separados e armazenados individualmente em variáveis do tipo *String*. O conteúdo destas variáveis é depois comparado com as informações recebidas a partir do *Intent* da



EnterPinActivity e com o comando da ordem de pagamento. No caso de esse conteúdo ser o esperado, o valor de uma variável booleana específica irá ser modificado para *true*, o que significa que o utilizador foi correctamente identificado e que será dada continuação ao processo de pagamento. Caso contrário, o processo será cancelado.

De seguida, novamente no *loop()* anteriormente mencionado é executado o código que conclui o processo de pagamento. A tabela 4.10 apresenta o referido código.

```
public class PayActivity extends IOIOActivity
{
    (...)

    class Looper extends BaseIOIOLooper
    {
        (...)

        @Override
        public void loop() throws ConnectionLostException
        {
            (...)

            if (done == true && receivedUserString == true)
            {
                if (printed == false)
                {
                    final SharedPreferences pref =
                        getSharedPreferences("BCCTerminalPrefs",
                            MODE_PRIVATE);
                    storedEntityName =
                        pref.getString("storedEntityName", "-");
                    storedEntityAccount =
                        pref.getString("storedEntityAccount", "-");
                    storedEntityCash = pref.getInt("storedEntityCash",
                        0);
                    Time time = new Time(Time.getCurrentTimezone());
                    time.setToNow();
                    date = time.monthDay + "/" + (time.month+1) + "/" +
                        time.year + " - " + time.format("%k:%M:%S");

                    try {
                        stream_out.write(("[" + storedEntityName +
                            "#" + storedEntityAccount + "#" +
                            String.valueOf(amount) + "#" + date +
                            "]" ).getBytes());
                        uart.close(); countdownTimer.cancel();
                        storedEntityCash = storedEntityCash +
                            amount;
                        SharedPreferences.Editor editor =
                            pref.edit();
                        editor.putInt("storedEntityCash", storedEntit
                            yCash); editor.commit();
                        runOnUiThread(new Runnable() {
                            public void run() {
                                new doPayment().execute();
                            }
                        });
                        printed == true;
                    } catch (IOException e) {}

                }
            }
        }
    }
}
```

Tabela 4.10. Conclusão do processo de pagamento (PayActivity).

Como se pode analisar pela tabela 4.10, os dados relativos à entidade vendedora são obtidos a par da data actual e são armazenados em variáveis temporárias, sendo logo de seguida enviados para o *smartphone* do utilizador via UART conjuntamente com a quantia inserida, permitindo deste modo que a factura do pagamento seja também gerada na carteira digital do utilizador. A figura 4.23 apresenta a trama enviada pelo terminal de pagamentos para o *smartphone* do utilizador, na qual também são utilizados caracteres especiais para delimitar os diversos campos.

[	Nome da entidade	#	Número de conta da entidade	#	Montante	#	Data	]
---	------------------	---	-----------------------------	---	----------	---	------	---

Figura 4.23. Trama enviada pelo terminal de pagamentos (conclusão).

Depois, a ligação de dados UART é encerrada, o temporizador é desactivado e o saldo da conta da entidade vendedora é incrementado com o valor da quantia inserida, ou seja, com o valor pago pelo utilizador. Como se pode observar, a obtenção dos dados e actualização do saldo da conta da entidade vendedora é feita recorrendo aos métodos associados à classe *SharedPreferences* e armazenando em variáveis temporárias os valores correspondentes. Para incrementar o saldo da entidade vendedora é efectuada uma operação de soma entre as variáveis do tipo *int* que contêm o saldo actual e a quantia inserida, sendo o resultado armazenado na primeira. Este valor é depois salvaguardado na respectiva variável da zona reservada do directório de dados da aplicação recorrendo novamente aos métodos associados à classe *SharedPreferences*.

Por fim, é invocada a *AsyncTask* que apresenta a factura no ecrã do terminal de pagamentos, isto é, os dados da entidade vendedora, os dados do utilizador, a quantia paga e a data de pagamento. Na tabela 4.11 é possível observar o seu respectivo código.

```
public class PayActivity extends IOIOActivity
{
    (...)

    private class doPayment extends AsyncTask<Void, Void, Boolean>
    {
        protected Boolean doInBackground(Void... params)
        {
            return receivedUserString;
        }

        protected void onPostExecute(Boolean receivedUserString)
        {
            if (receivedUserString == true)
            {
                loop_pay.setVisibility(View.INVISIBLE);
                processing_pay.setVisibility(View.INVISIBLE);
                pay_amount.setVisibility(View.VISIBLE);
                payed_amount.setVisibility(View.VISIBLE);
                payed_amount.setText("€ " + String.valueOf(amount));
                from_pay.setVisibility(View.VISIBLE);
                to_pay.setVisibility(View.VISIBLE);
                date_pay.setVisibility(View.VISIBLE);
                inserted_from_pay.setVisibility(View.VISIBLE);
                inserted_from_pay.setText(receivedUserName + "\nNIB " +
                    receivedUserAccount);
                inserted_to_pay.setVisibility(View.VISIBLE);
                inserted_to_pay.setText(storedEntityName + "\nNIB " +
                    storedEntityAccount);
            }
        }
    }
}
```

```

        inserted_date_pay.setVisibility(View.VISIBLE);
        inserted_date_pay.setText(date);
        processed_pay.setVisibility(View.VISIBLE);
        ok_button_pay.setVisibility(View.VISIBLE);
    }
    else
    {
        Toast toast2 = Toast.makeText(PayActivity.this, "Operação
        anulada", Toast.LENGTH_LONG);
        toast2.show();
        finish();
        overridePendingTransition(R.anim.fade_in, R.anim.fade_out);
    }
} (...)
}

```

Tabela 4.11. Apresentação da factura no ecrã do terminal (PayActivity).

Depois de apresentados os dados do pagamento, o processo é completamente finalizado ao ser clicado o *Button* implementado para o efeito, o qual invoca a *MainActivity* permitindo assim que novos pagamentos sejam efectuados.

Em última nota, é importante referir que foram incluídas no projecto as bibliotecas necessárias para a comunicação com a placa IOIO-OTG, nomeadamente, as bibliotecas IOIO-LibAccessory, IOIOLibAndroid e IOIOLibAndroidDevice. Neste sentido, foi também necessário incluir no ficheiro *AndroidManifest* da aplicação o código que se apresenta na tabela 4.12.

```

<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="pt.ua.bccterminal"
    android:versionCode="1"
    android:versionName="1.0" >

    <uses-sdk
        android:minSdkVersion="16"
        android:targetSdkVersion="20" />

    <uses-permission
        android:name="android.permission.INTERNET" />

    <uses-feature
        android:name="android.hardware.usb.host" />

    <application
        android:allowBackup="true"
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme" >

        <uses-library
            android:name="com.android.future.usb.accessory"
            android:required="false" />

        <activity
            android:name=".MainActivity"
            android:launchMode="singleTask"
            android:configChanges="orientation|keyboardHidden|screenSize"
            android:label="@string/app_name"
            android:theme="@style/FullscreenTheme" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
</manifest>

```

```
        </intent-filter>
        <intent-filter>
            <action android:name="android.hardware.usb.action.USB_ACCESSORY_ATTACHED"
                />
        </intent-filter>
        <intent-filter>
            <action android:name="android.hardware.usb.action.USB_DEVICE_ATTACHED" />
        </intent-filter>
        <meta-data
            android:name="android.hardware.usb.action.USB_ACCESSORY_ATTACHED"
            android:resource="@xml/accessory_filter" />
        <meta-data
            android:name="android.hardware.usb.action.USB_DEVICE_ATTACHED"
            android:resource="@xml/device_filter" />
    </activity>
    (...)
</application>
</manifest>
```

Tabela 4.12. AndroidManifest da aplicação BCC Terminal.

#### 4.4.6 BCC Wallet

De modo semelhante à secção anterior, na presente secção destacam-se os aspectos mais relevantes do *software* desenvolvido no que toca à aplicação BCC Wallet. A aplicação foi desenvolvida utilizando também o programa Eclipse ADT (versão 23.0.2) e foi testada utilizando igualmente um *smartphone* Samsung Galaxy S3 equipado com Android 4.3 (Jelly Bean).

Como foi possível observar na secção 4.3.3, a aplicação BCC Wallet é composta por três *Activities* e um *Service*, nomeadamente:

- MainActivity
- SettingsActivity
- PayActivity
- AppService

A MainActivity é a primeira *Activity* que é invocada logo após se abrir a aplicação e permite ao utilizador ver o seu saldo actual, ver uma lista com todos os pagamentos efectuados, bloquear/desbloquear a permissão para efectuar pagamentos, esvaziar a lista dos pagamentos e aceder à SettingsActivity. Por sua vez, esta última foi implementada de forma semelhante à da aplicação BCC Terminal, pelo que não se justifica novamente a sua explicação pois já foi feita na secção anterior. Apenas se faz menção ao facto de que para guardar, obter e apresentar os dados relativos ao utilizador também se usam os métodos associados à classe *SharedPreferences*, tal como na aplicação BCC Terminal.

A apresentação do saldo actual do utilizador é feita através de uma *TextView* cujo texto é obtido a partir de uma variável do tipo *String* na qual se armazenou previamente esse conteúdo recorrendo aos métodos associados à classe *SharedPreferences*.

O bloqueio ou desbloqueio da permissão para efectuar pagamentos é feito clicando no *ImageButton* implementado para o efeito, o qual invoca um método que basicamente activa ou desactiva o AppService. Este último é por sua vez um *Service* que, caso activo, corre continuamente em *background* independentemente de a aplicação estar visível ou não e tem como função aguardar e efectuar a comunicação com um terminal de pagamentos e processar todos

os novos pagamentos. Como tal, o mesmo será explicado mais adiante nesta secção. No caso de se pretender activar este *Service*, após o *ImageButton* ser clicado surgirá um *AlertDialog* solicitando a inserção do PIN do utilizador, aumentando-se assim a segurança da aplicação. A tabela 4.13 apresenta o código desenvolvido para activar ou desactivar o *AppService*.

```
public class MainActivity extends Activity
{
    public ImageButton secure_button;

    (...)

    public void secureButton(View v)
    {
        if (storedUserServiceState == false)
        {
            passwordSettingsDialog();
        }
        else
        {
            (...)
            stopService(new Intent(this, AppService.class));
            Toast toast = Toast.makeText(getBaseContext(), "BCC Wallet
            bloqueada", Toast.LENGTH_SHORT);
            toast.show();
        }
    }

    @SuppressWarnings("InflateParams")
    private void passwordSettingsDialog()
    {
        (...)
        AlertDialog.Builder passwordDialogBuilder = new
        AlertDialog.Builder(MainActivity.this,
        AlertDialog.THEME_DEVICE_DEFAULT_DARK)
        .setTitle("Inserir PIN actual:")
        .setView(dialogView)
        .setCancelable(true)
        .setPositiveButton("OK", new DialogInterface.OnClickListener()
        {
            public void onClick(DialogInterface dialog, int id1)
            {
                if ((edittext_dialog.getText().toString())
                .equals(storedUserPassword))
                {
                    (...)
                    startService(new Intent(getBaseContext(),
                    AppService.class));
                    Toast toast = Toast.makeText(getBaseContext(), "BCC
                    Wallet desbloqueada", Toast.LENGTH_SHORT);
                    toast.show();
                }
                else
                {
                    Toast toast = Toast.makeText(getBaseContext(), "PIN
                    incorrecto", Toast.LENGTH_SHORT);
                    toast.show();
                }
            }
        })
        .setNegativeButton("Cancelar", new DialogInterface.OnClickListener()
        {
            public void onClick(DialogInterface dialog, int id2)
            {
                dialog.dismiss();
            }
        })
    }
}
```

```

    });
    AlertDialog passwordDialog = passwordDialogBuilder.create();
    passwordDialog.show();
} (...)
}

```

Tabela 4.13. Activação/desactivação do AppService (MainActivity).

A lista dos pagamentos efectuados é implementada através de uma *ListView* cujas linhas são preenchidas (via *ArrayAdapter*) a partir de uma *ArrayList* contendo todos os dados de cada pagamento armazenados em formato *String* em cada uma das suas posições. Esses dados são por sua vez obtidos a partir de uma base de dados SQL interna à aplicação, a qual foi implementada para armazenar as facturas de cada pagamento (seguindo a estrutura apresentada na figura 4.16), permitindo que as mesmas sejam visualizadas após se clicarem as linhas correspondentes da *ListView*. Tal como o AppService, a base de dados será também explicada mais adiante nesta secção.

Então, para permitir a visualização das facturas implementou-se um método no qual ao ser clicada uma das linhas da *ListView*, este calcula o *id* do pagamento correspondente da base de dados a partir do índice da linha clicada e obtém os dados desse pagamento recorrendo a um método específico da classe da base de dados implementada (DBHelper), armazenando-os logo de seguida numa variável temporária (objecto da classe Payment cuja construção tem por base a estrutura da figura 4.16). Depois, é invocada a PayActivity passando-se um *Intent* contendo os dados obtidos (a partir dos métodos da classe Payment). A tabela 4.14 apresenta o código que foi desenvolvido neste âmbito.

```

public class MainActivity extends Activity
{
    public ListView list_view;
    private DBHelper db;
    private List<Payment> payments_list;
    private Payment payments_list;

    (...)

    @Override
    protected void onCreate(Bundle savedInstanceState)
    {
        (...)
        list_view = (ListView) findViewById(R.id.list_view);
        getData(); getList();
    }

    public void getList()
    {
        db = new DBHelper(this);
        payments_list = db.getAllPayments();
        if (payments_list.size() == 0) {
            list_view.setVisibility(View.INVISIBLE);
        }
        else {
            Collections.reverse(payments_list);
            String payments = payments_list.toString();
            String[] parts = payments.split("\n, ");
            ArrayList<String> payments_array = new ArrayList<String>();
            for (int i = 0; i < parts.length; i++)
            {
                parts[i] = parts[i].replace("[", ""); parts[i] =
                parts[i].replace("]", ""); parts[i] =

```

```

        parts[i].replace("\n", "");
        String[] part = parts[i].split("#");
        String all = part[1] + " #" + part[0] + "\n" + part[4] +
        "\n" + part[7] + "\n" + part[6] + " €";
        payments_array.add(all);
    }
    ArrayAdapter<String> adapter = new
    ArrayAdapter<String>(getBaseContext(),
    R.layout.custom_list, android.R.id.text1, payments_array);
    list_view.setAdapter(adapter);
    list_view.setVisibility(View.VISIBLE);
    list_view.setOnItemClickListener(new OnItemClickListener() {
    @Override
    public void onItemClick(AdapterView<?> parent, View view, int
    position, long id) {
        payment = db.getPayment(payments_list.size() - position);
        Intent intent = new Intent(getBaseContext(),
        PayActivity.class);
        intent.putExtra("storedUserName", payment.getUserName());
        intent.putExtra("storedUserAccount",
        payment.getUserAccount());
        intent.putExtra("receivedEntityName",
        payment.getEntityName());
        intent.putExtra("receivedEntityAccount",
        payment.getEntityAccount());
        intent.putExtra("receivedAmount", payment.getAmount());
        intent.putExtra("receivedDate", payment.getDate());
        startActivity(intent);
    }
    });
    }
    } (...)
}

```

Tabela 4.14. Lista dos pagamentos efectuados (MainActivity).

A lista dos pagamentos pode ser esvaziada clicando-se no *Button* correspondente, implementado na *Action Bar* da MainActivity. Este invoca uma *AsyncTask* que é executada em *background* e que apaga todas as entradas da base dados através do cálculo do tamanho da lista e do uso dos métodos presentes na classe da base de dados implementada. O código da tabela 4.15 mostra esta implementação.

```

public class MainActivity extends Activity
{
    public ListView list_view;
    private DBHelper db;
    private List<Payment> payments_list;
    private Payment payments_list;

    (...)
    @Override
    public boolean onOptionsItemSelected(MenuItems item)
    {
        switch (item.getItemId())
        {
            (...)
            case R.id.delete:
                if (payments_list.size() > 0)
                {
                    confirmationDialog();
                }
                else
                {
                    Toast toast = Toast.makeText(getBaseContext(),

```

```

        "Lista vazia", Toast.LENGTH_SHORT);
        toast.show();
    }
    return true;
}
return super.onOptionsItemSelected(item);
}

private void confirmationDialog()
{
    AlertDialog.Builder confirmationDialogBuilder = new
    AlertDialog.Builder(MainActivity.this,
    AlertDialog.THEME_DEVICE_DEFAULT_DARK)
    .setTitle("Apagar lista:")
    .setMessage("Tem a certeza?")
    .setCancelable(true)
    .setPositiveButton("Sim", new DialogInterface.OnClickListener()
    {
        public void onClick(DialogInterface dialog, int id1)
        {
            runOnUiThread(new Runnable() {
                public void run()
                {
                    new clearList().execute();
                }
            });
        }
    })
    .setNegativeButton("Não", new DialogInterface.OnClickListener()
    {
        public void onClick(DialogInterface dialog, int id2)
        {
            dialog.dismiss();
        }
    });
    AlertDialog confirmationDialog = confirmationDialogBuilder.create();
    confirmationDialog.show();
}

public class clearList extends AsyncTask<Void, Void, Boolean>
{
    protected Boolean doInBackground(Void... params)
    {
        while(payments_list.size() > 0)
        {
            db = new DBHelper(getBaseContext());
            payments_list = db.getAllPayments();
            if (payments_list.size() == 0) {}
            else
            {
                db.deleteBook(payments_list.get(0));
            }
        }
        return true;
    }
    protected void onPostExecute(Boolean var)
    {
        if (var == true)
        {
            list_view.setVisibility(View.INVISIBLE);
            Toast toast = Toast.makeText(getBaseContext(), "Lista
            vazia", Toast.LENGTH_SHORT); toast.show();
        }
    }
} (...)
```

Tabela 4.15. Esvaziamento da lista dos pagamentos efectuados (MainActivity).



A `PayActivity` é, tal como na aplicação BCC Terminal, a *Activity* que apresenta a factura de um pagamento. Esta pode ser invocada de duas formas. Ou pela selecção de uma linha da *ListView* da `MainActivity` (já explicada) ou a partir do `AppService` após ter sido concluído um pagamento. Em ambos os casos a `PayActivity` recebe um *Intent* e obtém os dados relativos ao pagamento em questão, armazenando-os em variáveis temporárias do tipo *String*. Por fim, é apresentada a informação nas diversas *TextViews* implementadas, conforme mostra o código da tabela 4.16.

```
public class PayActivity extends Activity
{
    (...)

    @Override
    protected void onCreate(Bundle savedInstanceState)
    {
        (...)
        Intent intent = getIntent();
        storedUserName = intent.getStringExtra("storedUserName");
        storedUserAccount = intent.getStringExtra("storedUserAccount");
        receivedEntityName = intent.getStringExtra("receivedEntityName");
        receivedEntityAccount = intent.getStringExtra("receivedEntityAccount");
        receivedAmount = intent.getStringExtra("receivedAmount");
        receivedDate = intent.getStringExtra("receivedDate");
        pay_amount = (TextView) findViewById(R.id.pay_amount);
        payed_amount = (TextView) findViewById(R.id.payed_amount);
        from_pay = (TextView) findViewById(R.id.from_pay);
        to_pay = (TextView) findViewById(R.id.to_pay);
        date_pay = (TextView) findViewById(R.id.date_pay);
        inserted_from_pay = (TextView) findViewById(R.id.inserted_from_pay);
        inserted_to_pay = (TextView) findViewById(R.id.inserted_to_pay);
        inserted_date_pay = (TextView) findViewById(R.id.inserted_date_pay);
        processed_pay = (TextView) findViewById(R.id.processed_pay);
        ok_button_pay = (Button) findViewById(R.id.ok_button_pay);
        payed_amount.setText("€ " + String.valueOf(receivedAmount));
        inserted_from_pay.setText(storedUserName + "\nNIB " + storedUserAccount);
        inserted_to_pay.setText(receivedEntityName + "\nNIB " +
            receivedEntityAccount);
        inserted_date_pay.setText(receivedDate);
    }

    public void okButton(View v)
    {
        NavUtils.navigateUpFromSameTask(this);
    } (...)
}
```

Tabela 4.16. Apresentação da factura de um pagamento (`PayActivity`).

Feita a análise de cada *Activity* implementada, a partir deste ponto irá proceder-se à análise do *Service* da aplicação BCC Wallet.

O `AppService`, como já foi referido, é um *Service* que foi implementado para realizar a comunicação com um terminal (neste caso emulado por outro *smartphone*) e processar os pagamentos. Este é executado em *background* independentemente de a aplicação BCC Wallet se encontrar visível ou não no ecrã do *smartphone* do utilizador, o que permite que este último efectue pagamentos comodamente sem ter que abrir a aplicação ou manusear o *smartphone*, podendo deixá-lo no seu bolso, por exemplo. De modo semelhante à aplicação BCC Terminal, neste *Service* também se implementou um *loop()* contendo as operações necessárias ao processo do pagamento. Como tal, a par da inicialização de todas as variáveis é necessário primeiro

configurar a trama de dados UART, configuração esta que é feita utilizando novamente o código apresentado na tabela 4.5. O restante código do AppService relativo ao processamento dos pagamentos será explicado de seguida.

Primeiro, o *Service* executa o *loop()* implementado e, no caso de a variável booleana que permite que a aplicação efectue pagamentos ou não estiver como *true*, então a aplicação pode efectuá-los e, logo de seguida, o *Service* irá aguardar pela comunicação de um terminal a qualquer momento como ilustra o código da tabela 4.17.

```
public class AppService extends IOIOService
{
    public int countdown_time = 60;
    public Boolean payments_allowed = true;
    public CountDownTimer countDownTimer;

    (...)
    @Override
    protected IOIOLooper createIOIOLooper() {return new BaseIOIOLooper() {

    (...)
    private int i = 0,

    @Override
    public void loop() throws ConnectionLostException, InterruptedException {

        Thread.sleep(100);
        if(payments_allowed == true)
        {
            if(i < 6) // if pay command has not been sent
            {
                if(received_char == '*')
                {
                    final SharedPreferences pref =
                        getSharedPreferences("BCCWalletPrefs", MODE_PRIVATE);
                    storedUserName = pref.getString("storedUserName", "-");
                    storedUserAccount = pref.getString("storedUserAccount", "-");
                    storedUserPassword = pref.getString("storedUserPassword",
                        "1234");

                    try {
                        stream_out.write(("[" + storedUserName + "#" +
                            storedUserAccount + "]" ).getBytes());
                    } catch (IOException e) {}

                    payment_started = false; confirmation_done = true;
                    received_pin = ""; received_string = "";
                    receivedEntityName = ""; receivedEntityAccount = "";
                    receivedAmount = ""; receivedDate = "";
                    amount = 0; i = 0;

                }

                try {
                    received_char = (char) stream_in.read();
                } catch (IOException e) {}

                if (confirmation_done == true && payment_started == true && i < 5)
                {
                    received_pin = received_pin + received_char;
                    i++;
                }

                if (confirmation_done == true && i == 0 && received_char == '@')
                {
                    payment_started = true;
                    received_pin = "";
                }
            }
        }
    }
}
```

```

        i++;
    }
    if (i == 5) // if pin has been read
    {
        if (received_pin.equals(storedUserPassword))
        {
            try {
                stream_out.write(("[" + storedUserName + "#"
                    + storedUserAccount + "#" + "PAY" +
                    "]" ).getBytes());
                i++;
            } catch (IOException e) {}
        }
        else // if pin is incorrect
        {
            try {
                stream_out.write('%'); // abort
            } catch (IOException e) {}
            i = 0;
            confirmation_done = false; payment_started = false;
        }
    }
    } (...)
}
else {}
}};}}

```

Tabela 4.17. Inicialização do processo de pagamento (AppService).

Pela tabela 4.17 é possível observar que as várias fases do processo do pagamento são executadas consoante a modificação do estado de determinadas variáveis booleanas e a incrementação de uma variável específica (tipo *int* inicializada a zero). Essas alterações são feitas sempre que uma das fases é executada com sucesso, caso contrário, todas as variáveis envolvidas são reinicializadas de modo a que seja necessário passar novamente por todas as fases do pagamento na próxima tentativa de comunicação.

No caso de o *smartphone* do utilizador ter recebido o carácter especial enviado pelo terminal de pagamentos, então o processo do pagamento é efectivamente iniciado obtendo-se e armazenando em variáveis temporárias tipo *String* os dados do utilizador através dos métodos associados à classe *SharedPreferences*. Essa informação é enviada logo de seguida através da ligação de dados UART numa trama igual à da figura 4.20. Depois, o *smartphone* do utilizador aguarda pela resposta do terminal de pagamentos, a qual consiste numa trama contendo um carácter especial seguido pelo PIN introduzido pelo utilizador, conforme ilustra a figura 4.21. O PIN recebido é comparado com o PIN armazenado no *smartphone* do utilizador e, caso esteja correcto, este último envia a trama apresentada na figura 4.22, indicando ao terminal de pagamentos que o pagamento deverá ser efectuado. Caso contrário, o mesmo envia para o terminal um carácter especial indicando que o pagamento deverá ser abortado.

Feita a identificação do utilizador e enviado o comando da ordem de pagamento, o contador é incrementado de modo a ser executada a última fase do processo de pagamento, a qual consiste no decremento do saldo do utilizador e na recepção da factura do pagamento. Esta fase é representada pelo código da tabela 4.18.

```

public class AppService extends IOIOService
{
    public int countdown_time = 60;
    public Boolean payments_allowed = true;
    public CountDownTimer countDownTimer;
}

```

```
(...)

@Override
protected IOIOLooper createIOIOLooper() {return new BaseIOIOLooper() {

(...)
private int i = 0,

@Override
public void loop() throws ConnectionLostException, InterruptedException {

Thread.sleep(100);
if(payments_allowed == true)
{
    (...)
    if(i == 6) // if pay command has been sent
    {
        try {
            received_char = (char) stream_in.read();
        } catch (IOException e) {}

        if (received_char == '*')
        {
            i = 0; confirmation_done = false; payment_started = false;
            start_string = false; end_string = false;
        }
        if (received_char == '[')
            start_string = true;
        if (start_string == true)
            received_string = received_string + received_char;
        if (received_char == ']')
            end_string = true;
        if (start_string == true && end_string == true) {
            if (received_string.contains("[") && received_string.contains("#")
            && received_string.contains("]")) {
                received_string = received_string.replace("[", "");
                received_string = received_string.replace("]", "");
                String[] parts = received_string.split("#");
                receivedEntityName = parts[0];
                receivedEntityAccount = parts[1];
                receivedAmount = parts[2];
                receivedDate = parts[3];
            }

            // decrement user cash
            final SharedPreferences pref =
            getSharedPreferences("UAWalletPrefs", MODE_PRIVATE);
            storedUserCash = pref.getInt("storedUserCash", 0);
            amount = Integer.parseInt(receivedAmount.replaceAll("[\\D]", ""));
            storedUserCash = storedUserCash - amount;
            SharedPreferences.Editor editor = pref.edit();
            editor.putInt("storedUserCash", storedUserCash);
            editor.commit();

            // notification
            NotificationManager nm = (NotificationManager)
            getSystemService(NOTIFICATION_SERVICE);
            Notification notification = new
            Notification(R.drawable.ic_launcher, "Pagamento efetuado",
            System.currentTimeMillis());
            Intent intent = new Intent(getBaseContext(), PayActivity.class);
            intent.putExtra("storedUserName", storedUserName);
            intent.putExtra("storedUserAccount", storedUserAccount);
            intent.putExtra("receivedEntityName", receivedEntityName);
            intent.putExtra("receivedEntityAccount", receivedEntityAccount);
            intent.putExtra("receivedAmount", receivedAmount);
            intent.putExtra("receivedDate", receivedDate);
            notification.setLatestEventInfo(getBaseContext(), "Pagamento
```



```
public class Payment
{
    private String id;
    private String title;
    private String user_name;
    private String user_account;
    private String entity_name;
    private String entity_account;
    private String amount;
    private String date;

    public Payment() {}

    public Payment(String title, String user_name, String user_account, String
entity_name, String entity_account, String amount, String date) {
        super();
        this.title = title;
        this.user_name = user_name;
        this.user_account = user_account;
        this.entity_name = entity_name;
        this.entity_account = entity_account;
        this.amount = amount;
        this.date = date;
    }

    public int getId() {return id;}
    public void setId(int id) {this.id = id;}
    public String getTitle() {return title;}
    public void setTitle(String title) {this.title = title;}
    public String getUsername() {return user_name;}
    public void setUsername(String user_name) {this.user_name = user_name;}
    public String getUserAccount() {return user_account;}
    public void setUserAccount(String user_account) {this.user_account = user_account;}
    public String getEntityName() {return entity_name;}
    public void setEntityName(String entity_name) {this.entity_name = entity_name;}
    public String getEntityAccount() {return entity_account;}
    public void setEntityAccount(String entity_account) {this.entity_account =
entity_account;}
    public String getAmount() {return amount;}
    public void setAmount(String amount) {this.amount = amount;}
    public String getDate() {return date;}
    public void setDate(String date) {this.date = date;}

    @Override
    public String toString() {
        return "[" + id + "#" + title + "#" + user_name + "#" + user_account + "#" +
entity_name + "#" + entity_account + "#" + amount + "#" + date + "]\n";
    }
}
```

Tabela 4.19. Classe Payment.

A base de dados, designada por PaymentsDatabase, foi implementada recorrendo à classe *SQLiteOpenHelper*, a qual fornece métodos específicos para a criação da base de dados e permite que esta seja armazenada numa zona reservada do directório de dados da aplicação. Então, criou-se uma subclasse designada por DBHelper que herda a classe *SQLiteOpenHelper* e que constrói a base de dados SQL segundo a estrutura definida para os pagamentos. Nesta classe estão também implementados os métodos que permitem adicionar novos pagamentos à base de dados, obter os já existentes ou apagar os mesmos, como se pode observar no código da tabela 4.20.

```

public class DBHelper extends SQLiteOpenHelper
{
    public static final String TABLE_NAME = "PaymentsTable";
    public static final String KEY_ID = "id";
    public static final String KEY_TITLE = "title";
    public static final String KEY_USERNAME = "user_name";
    public static final String KEY_USERACCOUNT = "user_account";
    public static final String KEY_ENTITYNAME = "entity_name";
    public static final String KEY_ENTITYACCOUNT = "entity_account";
    public static final String KEY_AMOUNT = "amount";
    public static final String KEY_DATE = "date";
    private static final String[] COLUMNS = {KEY_ID, KEY_TITLE, KEY_USERNAME,
        KEY_USERACCOUNT, KEY_ENTITYNAME, KEY_ENTITYACCOUNT, KEY_AMOUNT, KEY_DATE};
    private static final String TEXT_TYPE = " TEXT";
    private static final String COMMA_SEP = ",";
    private static final String DATABASE_CREATE =
        "CREATE TABLE " + TABLE_NAME + " (" + KEY_ID + " INTEGER PRIMARY KEY, " +
        KEY_TITLE + TEXT_TYPE + COMMA_SEP + KEY_USERNAME + TEXT_TYPE + COMMA_SEP +
        KEY_USERACCOUNT + TEXT_TYPE + COMMA_SEP + KEY_ENTITYNAME + TEXT_TYPE +
        COMMA_SEP + KEY_ENTITYACCOUNT + TEXT_TYPE + COMMA_SEP + KEY_AMOUNT +
        TEXT_TYPE + COMMA_SEP + KEY_DATE + TEXT_TYPE + " )";
    private static final String DATABASE_DELETE = "DROP TABLE IF EXISTS " +
        TABLE_NAME;
    private static final String DATABASE_NAME = "PaymentsDatabase.db";
    private static final int DATABASE_VERSION = 1;

    public DBHelper(Context context) {
        super(context, DATABASE_NAME, null, DATABASE_VERSION);
    }

    @Override
    public void onCreate(SQLiteDatabase database) {
        database.execSQL(DATABASE_CREATE);
    }

    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
        db.execSQL(DATABASE_DELETE);
        onCreate(db);
    }

    // add payment
    public void addPayment(Payment payment) {
        SQLiteDatabase db = this.getWritableDatabase();
        ContentValues values = new ContentValues();
        values.put(KEY_TITLE, payment.getTitle());
        values.put(KEY_USERNAME, payment.getUserName());
        values.put(KEY_USERACCOUNT, payment.getUserAccount());
        values.put(KEY_ENTITYNAME, payment.getEntityName());
        values.put(KEY_ENTITYACCOUNT, payment.getEntityAccount());
        values.put(KEY_AMOUNT, payment.getAmount());
        values.put(KEY_DATE, payment.getDate());
        db.insert(TABLE_NAME, null, values);
        db.close();
    }

    // get single payment
    public Payment getPayment(int id) {
        SQLiteDatabase db = this.getReadableDatabase();
        Cursor cursor = db.query(TABLE_NAME, COLUMNS, " id = ?", new String[]
            {String.valueOf(id)}, null, null, null, null);
        if (cursor != null)
            cursor.moveToFirst();
        Payment payment = new Payment();
        payment.setId(Integer.parseInt(cursor.getString(0)));
        payment.setTitle(cursor.getString(1));
        payment.setUserName(cursor.getString(2));
        payment.setUserAccount(cursor.getString(3));
    }
}

```

```

        payment.setEntityName(cursor.getString(4));
        payment.setEntityAccount(cursor.getString(5));
        payment.setAmount(cursor.getString(6));
        payment.setDate(cursor.getString(7));
        return payment;
    }

    // get all payments
    public List<Payment> getAllPayments() {
        List<Payment> payments = new LinkedList<Payment>();
        String query = "SELECT * FROM " + TABLE_NAME;
        SQLiteDatabase db = this.getWritableDatabase();
        Cursor cursor = db.rawQuery(query, null);
        Payment payment = null;
        if (cursor.moveToFirst()) {
            do {
                payment = new Payment();
                payment.setId(Integer.parseInt(cursor.getString(0)));
                payment.setTitle(cursor.getString(1));
                payment.setUserName(cursor.getString(2));
                payment.setUserAccount(cursor.getString(3));
                payment.setEntityName(cursor.getString(4));
                payment.setEntityAccount(cursor.getString(5));
                payment.setAmount(cursor.getString(6));
                payment.setDate(cursor.getString(7));
                payments.add(payment);
            } while (cursor.moveToNext());
        }
        return payments;
    }

    // delete single payment
    public void deleteBook(Payment payment) {
        SQLiteDatabase db = this.getWritableDatabase();
        db.delete(TABLE_NAME, KEY_ID+" = ?", new String[]
        {String.valueOf(payment.getId())});
        db.close();
    }
}

```

Tabela 4.20. Classe DBHelper.

Em última nota, tal como para a aplicação BCC Terminal foram também incluídas no projecto as bibliotecas necessárias para a comunicação com a placa IOIO-OTG (IOIOLib-Accessory, IOIOLibAndroid e IOIOLibAndroidDevice). De igual modo, foi necessário incluir no ficheiro AndroidManifest da aplicação o código que se apresenta na tabela 4.21.

```

<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="pt.ua.bccwallet"
    android:versionCode="1"
    android:versionName="1.0" >

    <uses-sdk
        android:minSdkVersion="16"
        android:targetSdkVersion="20" />

    <uses-permission
        android:name="android.permission.INTERNET" />

    <uses-feature
        android:name="android.hardware.usb.host" />

    <application

```



```

        android:allowBackup="true"
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme" >

        <uses-library
            android:name="com.android.future.usb.accessory"
            android:required="false" />

        <service android:name=".AppService" />

        <activity
            android:name=".MainActivity"
            android:launchMode="singleTask"
            android:configChanges="orientation|keyboardHidden|screenSize"
            android:label="@string/app_name"
            android:theme="@style/FullscreenTheme" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
            <intent-filter>
                <action android:name="android.hardware.usb.action.USB_ACCESSORY_ATTACHED"
                    />
            </intent-filter>
            <intent-filter>
                <action android:name="android.hardware.usb.action.USB_DEVICE_ATTACHED" />
            </intent-filter>
            <meta-data
                android:name="android.hardware.usb.action.USB_ACCESSORY_ATTACHED"
                android:resource="@xml/accessory_filter" />
            <meta-data
                android:name="android.hardware.usb.action.USB_DEVICE_ATTACHED"
                android:resource="@xml/device_filter" />
        </activity>
        (...)
    </application>
</manifest>

```

Tabela 4.21. AndroidManifest da aplicação BCC Wallet.

## 4.5 Considerações finais

Ao finalizar este capítulo é necessário ter em consideração certos aspectos do sistema de pagamento desenvolvido relativamente à sua implementação e segurança.

É importante frisar que o sistema implementado trata-se de um protótipo e por isso o mesmo apresenta algumas implementações que apenas existem para efeitos de teste, pelo que na sua versão final estas deveriam ser removidas ou alteradas. No entanto, as mesmas foram realizadas por motivos de simplificação do sistema e por permitirem testá-lo mais facilmente, dado o curto espaço de tempo que existiu para a sua implementação. Por outro lado, algumas implementações serviram como soluções alternativas a outras que por sua vez seriam impraticáveis pelas razões já mencionadas em secções anteriores. Como tal, realçam-se os aspectos mais relevantes que derivaram dessas implementações:

- A possibilidade da introdução do saldo da entidade vendedora ou do utilizador nas janelas das preferências das aplicações BCC Terminal e BCC Wallet, respectivamente. Do mesmo modo, a introdução dos restantes dados sem existir qualquer verificação de que são válidos (por meio de comunicação com uma entidade bancária por exemplo).

- O facto de os pagamentos estarem restringidos apenas a quantias com números naturais. No entanto, esta restrição pode deixar de existir após serem efectuados os respectivos ajustes no *software* de ambas as aplicações BCC Terminal e BCC Wallet.
- Considerou-se que o utilizador pode efectuar pagamentos mesmo que não tenha saldo suficiente para a quantia inserida, pelo que o seu saldo após ser decrementado ficará negativo nesse caso.
- A não criação de uma base de dados dos pagamentos efectuados para a aplicação BCC Terminal. Para tal, apenas teria de se repetir de certo modo a implementação realizada para a aplicação BCC Wallet.
- A impossibilidade de identificar o utilizador através da sua impressão digital (ou outra via) em vez do seu PIN apenas. Uma solução seria por exemplo utilizar um *smartphone* equipado com um módulo de leitura de impressões digitais e desenvolver o código para a referida implementação.
- A utilização de alguns componentes e circuitos externos aos *smartphones*.

Em relação à segurança do sistema implementado, pode afirmar-se que o mesmo possui algumas características que o tornam mais seguro mas também possui outras que o tornam menos seguro. Como tal, é possível enumerar os seguintes aspectos:

- O facto de a comunicação ser realizada através de acoplamento capacitivo ao corpo humano torna o sistema só por si mais seguro.
- A imposição de intervalos de tempo pré-definidos entre pagamentos consecutivos aumenta a segurança do sistema.
- A necessidade de identificar o utilizador duas vezes (com solicitação de PIN na segunda vez) e a de ter que se passar por todas as fases do processo do pagamento mesmo depois de ter ocorrido algum erro tornam o sistema mais seguro.
- A necessidade de introdução de *password* ou PIN para visualizar determinados conteúdos das aplicações do sistema aumenta a sua segurança.
- A não utilização de algum algoritmo de encriptação (por exemplo) na comunicação do sistema torna-o menos seguro.
- A eventual replicação de um terminal de pagamentos fraudulento por parte de uma entidade por sua vez também fraudulenta é sempre uma possibilidade, o que torna o sistema menos seguro neste aspecto. Contudo, essa situação poderia ser detectada se (por exemplo) fosse implementada uma base de dados contendo entidades vendedoras certificadas com a qual a aplicação BCC Wallet comunicaria durante o processo de pagamento. Neste caso, o pagamento seria automaticamente cancelado.

Em conclusão, e tendo em vista novamente o facto de o sistema implementado se encontrar ainda em fase de protótipo, desenvolveu-se um sistema funcional que poderá efectuar pagamentos através do corpo humano ao integrar na sua arquitectura o *transceiver* do sistema de comunicação também desenvolvido no âmbito deste projecto de dissertação. No entanto, este sistema de pagamento poderá ser melhorado no futuro e novas implementações poderão ser adicionadas tanto a nível de *hardware* como de *software*.

# Capítulo 5

## Resultados

### 5.1 Sumário

Neste capítulo são apresentadas todas as experiências realizadas com os sistemas de comunicação e pagamento implementados e são analisados os resultados obtidos.

A primeira secção é dedicada apenas ao sistema de comunicação e é onde se medem as formas de onda dos sinais mais relevantes e os seus respectivos espectros. São também determinados o alcance e consumo de energia do *transceiver*. A secção seguinte é dedicada ao sistema de pagamento e é onde se apresentam as simulações e os testes realizados com este no caso em que não é utilizado o *transceiver* do sistema de comunicação e no caso em que o mesmo é utilizado.

### 5.2 Sistema de comunicação

#### 5.2.1 Transceiver implementado

Na figura 5.1 é possível observar o *transceiver* implementado no âmbito deste projecto de dissertação de mestrado.

Todos os componentes presentes no *transceiver* foram soldados manualmente e o eléctrodo foi coberto com fita isolante de forma a garantir que não existiria contacto directo com o corpo humano durante as medições dos sinais relevantes. O *transceiver* assemblado possui dimensões de 5 x 5 x 1 cm, as quais são efectivamente menores que as das suas versões anteriores. Mais uma vez, refere-se que estas dimensões poderiam ser mais reduzidas se tivessem sido considerados os factores já mencionados na secção 3.8, algo que deverá ser feito apenas numa versão final do *transceiver*.

O eléctrodo construído consiste numa placa de cobre de dimensões semelhantes às da PCB do *transceiver*. O mesmo foi soldado directamente no ponto de teste correspondente à saída do amplificador sintonizado do emissor/receptor. O objectivo desta implementação é que o eléctrodo assente horizontalmente sobre a PCB do *transceiver* de modo a que este último obtenha uma estrutura compacta e uniforme. Assim, o acoplamento capacitivo entre o corpo humano e o eléctrodo dependerá apenas da posição e orientação do *transceiver*.

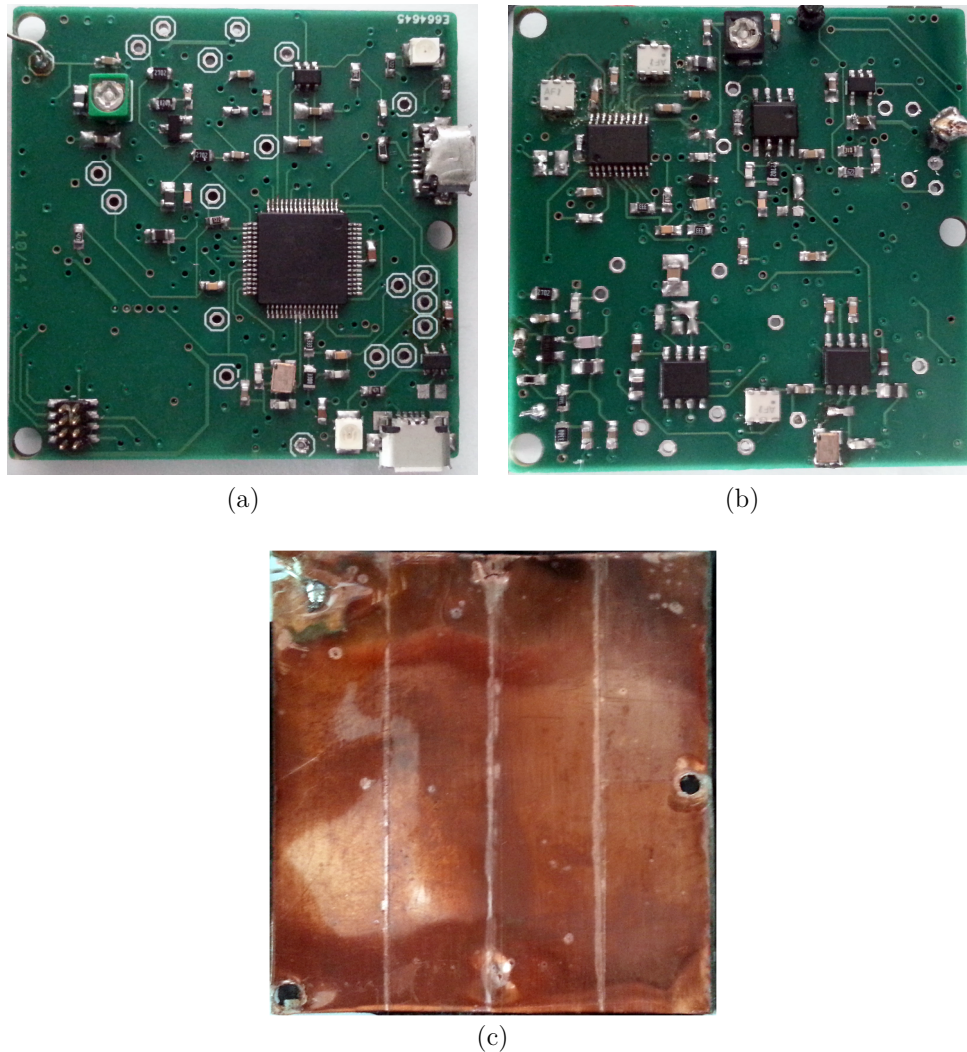


Figura 5.1. (a) Camada superior (*top layer*) do *transceiver*. (b) Camada inferior (*bottom layer*) do *transceiver*. (c) Eléctrodo do *transceiver*.

### 5.2.2 Bloco emissor do transceiver

Nesta secção apresentam-se os resultados obtidos para o bloco emissor do *transceiver*. Relembra-se que este bloco é constituído pelos seguintes andares:

- Modulador FSK (gera o sinal BFSK)
- IF *up-conversion mixer* (gera o sinal IF BFSK)
- Filtro passa-banda (filtra o sinal IF BFSK)
- RF *up-conversion mixer* (gera o sinal RF BFSK)
- Amplificador sintonizado (amplifica o sinal RF BFSK a transmitir)

A figura 5.2 apresenta o sinal da trama de dados UART a transmitir (neste caso uma sequência binária genérica com um *baudrate* de 30 Kbps) e o sinal BFSK gerado pelo modulador FSK a par do seu respectivo espectro. Repare-se que os símbolos 0 e 1 do sinal BFSK correspondem às frequências de 666 KHz e 800 KHz, respectivamente.

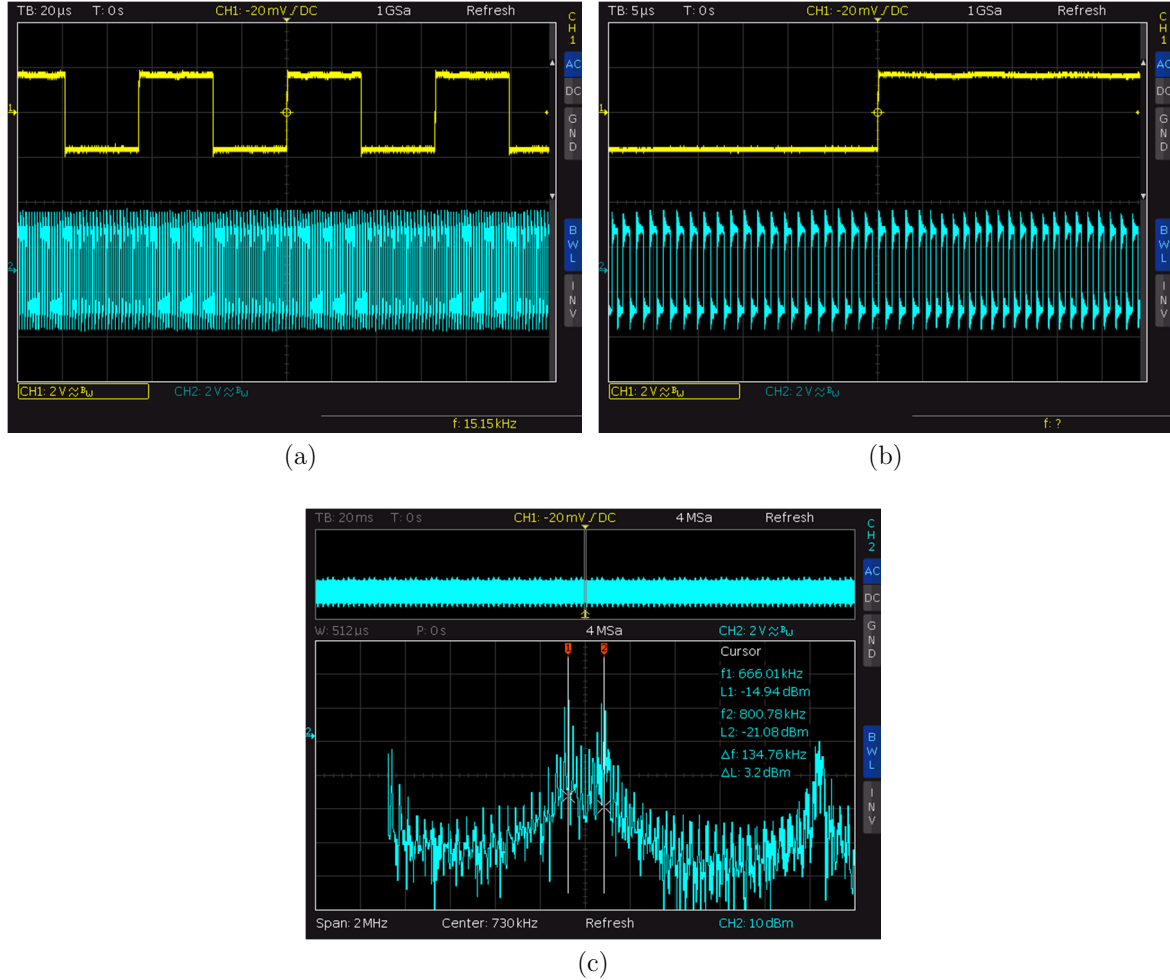


Figura 5.2. (a) Sinais à entrada e saída do modulador FSK. (b) Sinais à entrada e saída do modulador FSK (sinais ampliados). (c) Espectro do sinal modulado.

A figura 5.3 apresenta o sinal à entrada do IF *up-conversion mixer* e o sinal IF BFSK gerado à saída a par do seu respectivo espectro. É possível observar que o sinal BFSK foi transladado para a frequência intermédia de 10.7 MHz e que os símbolos 0 e 1 passaram a corresponder às frequências de 10.666 MHz e 10.800 MHz, respectivamente. Em pequena nota, relembra-se que devido ao elevado ganho do *mixer* utilizado (NXP SA602A), o sinal BFSK é atenuado e filtrado antes de ser aplicado à entrada do *mixer*, como se pode observar.

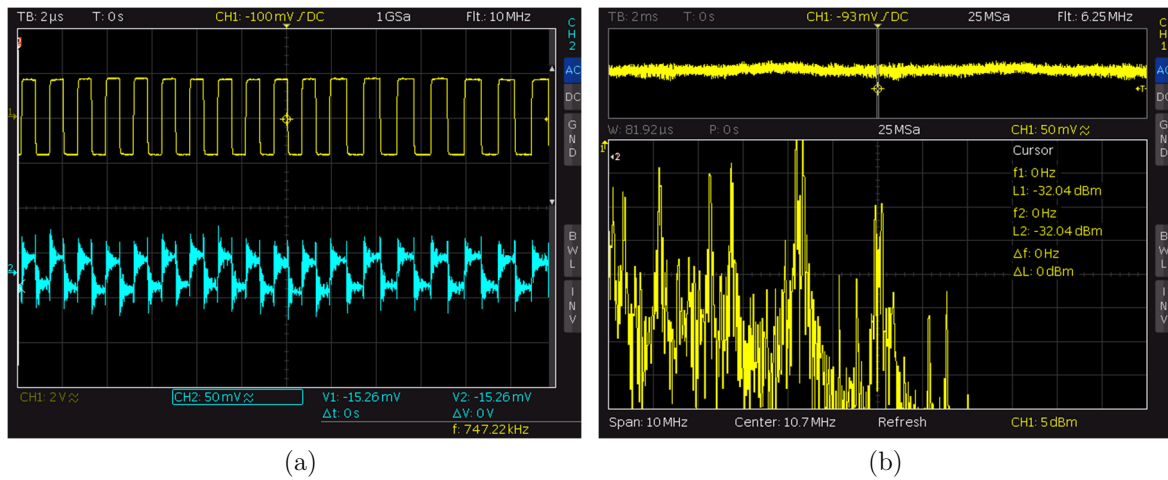


Figura 5.3. (a) Sinal à entrada do IF *up-conversion mixer*. (b) Espectro do sinal à saída do IF *up-conversion mixer*.

A figura 5.4 apresenta o sinal IF BFSK filtrado pelo filtro passa-banda e o seu respectivo espectro. Observa-se que o filtro atenua bastante as frequências fora da gama de interesse do sinal IF BFSK. Verificou-se também na prática que o sinal depois de filtrado passa a aproximar-se mais de uma sinusóide. Tal ocorre devido ao facto de o filtro deixar passar apenas a frequência fundamental do sinal e as suas frequências correspondentes aos símbolos 0 e 1, afastando-o por isso de um sinal quadrado cujo espectro é idealmente infinito.

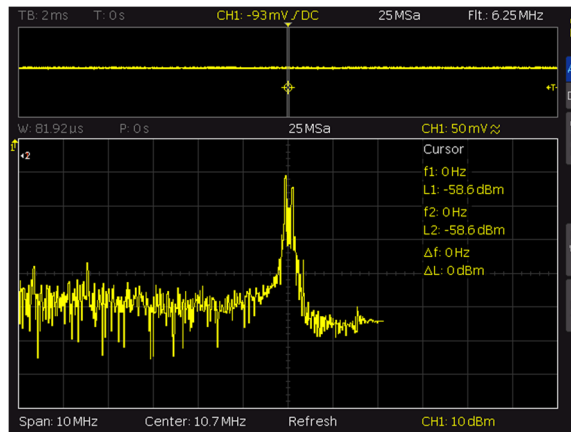
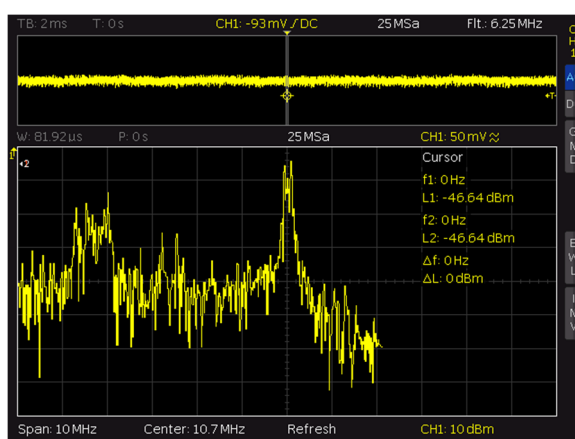


Figura 5.4. Espectro do sinal à saída do filtro passa-banda do emissor.

A figura 5.5 apresenta o sinal RF BFSK gerado pelo RF *up-conversion mixer* e o seu respectivo espectro. Observa-se que neste andar são introduzidas componentes de frequência indesejadas no sinal, o que por sua vez tem como consequência o aparecimento de distorção neste último.


 Figura 5.5. Espectro do sinal à saída do RF *up-conversion mixer*.

A figura 5.6 apresenta o sinal transmitido (sinal RF BFSK amplificado) e o seu respectivo espectro. Verifica-se que o sinal possui uma amplitude considerável e inclusive maior que a tensão de alimentação do *transceiver*, algo que se deve ao efeito ressonante da malha LC do amplificador sintonizado pois a mesma apresenta uma impedância elevadíssima à frequência intermédia do sinal transmitido. Tal como mencionado em [36], convém referir que o próprio acto de medição do sinal afecta a sua resposta, pois ao se utilizar uma ponta de prova de um osciloscópio introduz-se por sua vez uma capacidade adicional na malha LC do amplificador sintonizado, pelo que a sua resposta poderá ser ligeiramente diferente da apresentada.

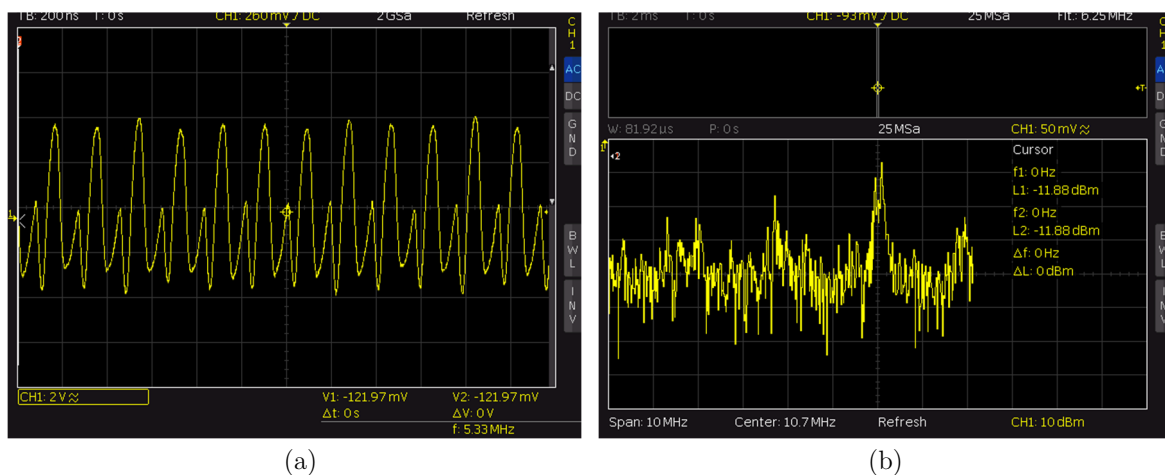


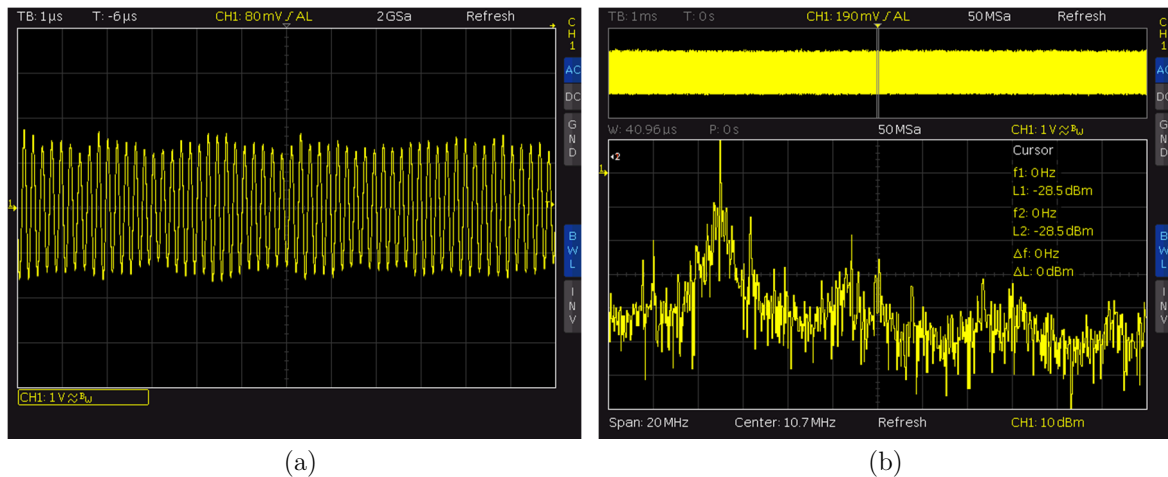
Figura 5.6. (a) Sinal à saída do amplificador sintonizado do emissor. (b) Espectro do sinal à saída do amplificador sintonizado do emissor.

### 5.2.3 Bloco receptor do transceiver

Nesta seção apresentam-se os resultados obtidos para o bloco receptor do *transceiver*. Relembra-se que este bloco é constituído pelos seguintes andares:

- Amplificador sintonizado (amplifica o sinal RF BFSK recebido)
- RF *down-conversion mixer* (gera o sinal IF BFSK)
- Filtro passa-banda (filtra o sinal IF BFSK)
- Amplificador FI (amplifica o sinal IF BFSK)
- Filtro passa-banda (filtra o sinal IF BFSK)
- Limitador (limita o sinal IF BFSK)
- Desmodulador FSK (gera o sinal desmodulado)
- Amplificador de banda-base (amplifica o sinal desmodulado)
- Comparador (gera o sinal digital binário a receber)

A figura 5.7 apresenta os sinais à entrada e saída do amplificador sintonizado (sinal RF BFSK recebido e sinal RF BFSK amplificado) a par dos seus respectivos espectros. É possível observar que são introduzidas componentes de frequência indesejadas no sinal recebido e que o mesmo é depois amplificado e filtrado pelo amplificador sintonizado.





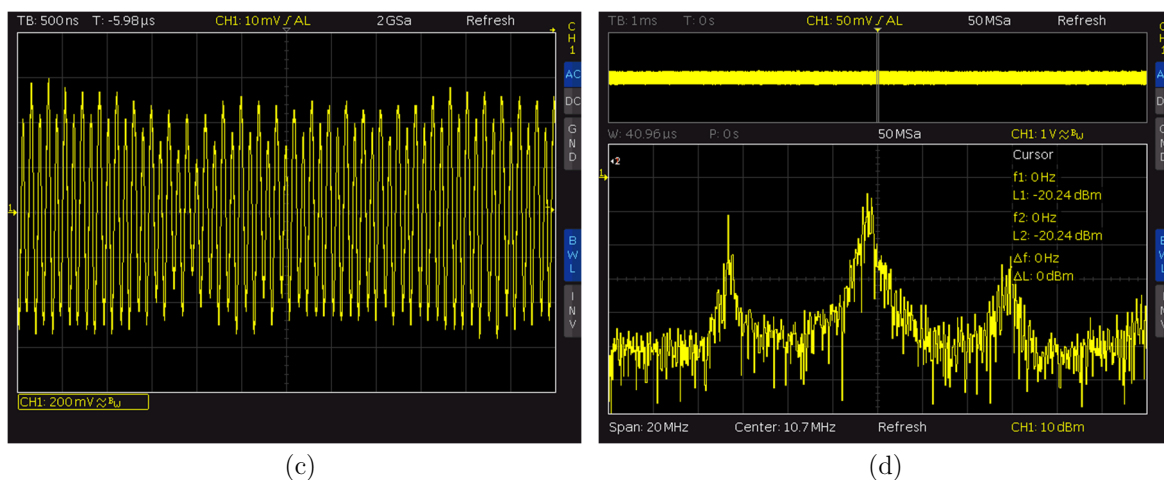


Figura 5.7. (a) Sinal à entrada do amplificador sintonizado do receptor. (b) Espectro do sinal à entrada do amplificador sintonizado do receptor. (c) Sinal à saída do amplificador sintonizado do receptor. (d) Espectro do sinal à saída do amplificador sintonizado do receptor.

A figura 5.8 apresenta o sinal gerado pelo RF *down-conversion mixer* e o seu respectivo espectro. Observa-se que o sinal IF BFSK gerado apresenta alguma distorção devido aos harmónicos introduzidos neste andar.

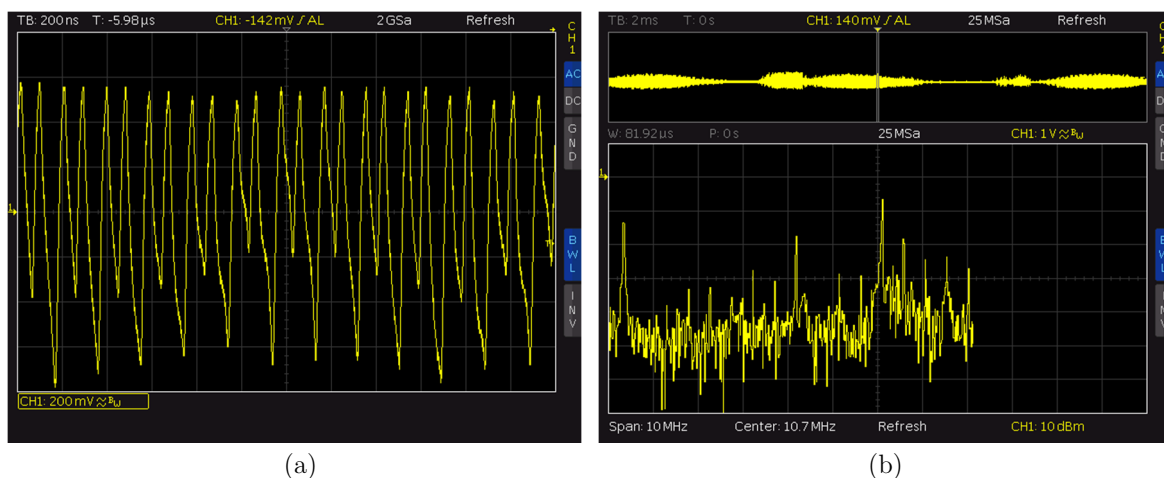


Figura 5.8. (a) Sinal à saída do RF *down-conversion mixer*. (b) Espectro do sinal à saída do RF *down-conversion mixer*.

A figura 5.9 apresenta o sinal IF BFSK à entrada e saída do amplificador FI e o seu respectivo espectro. À entrada do amplificador FI o sinal apresenta claras melhorias pois o filtro passa-banda a montante atenua significativamente todas as componentes de frequência fora da gama de interesse. Contudo, devido à filtragem o sinal é bastante atenuado. Depois, à saída do amplificador FI obtém-se um sinal com maior amplitude mas também com alguma distorção, novamente devido à amplificação de harmónicos indesejados.

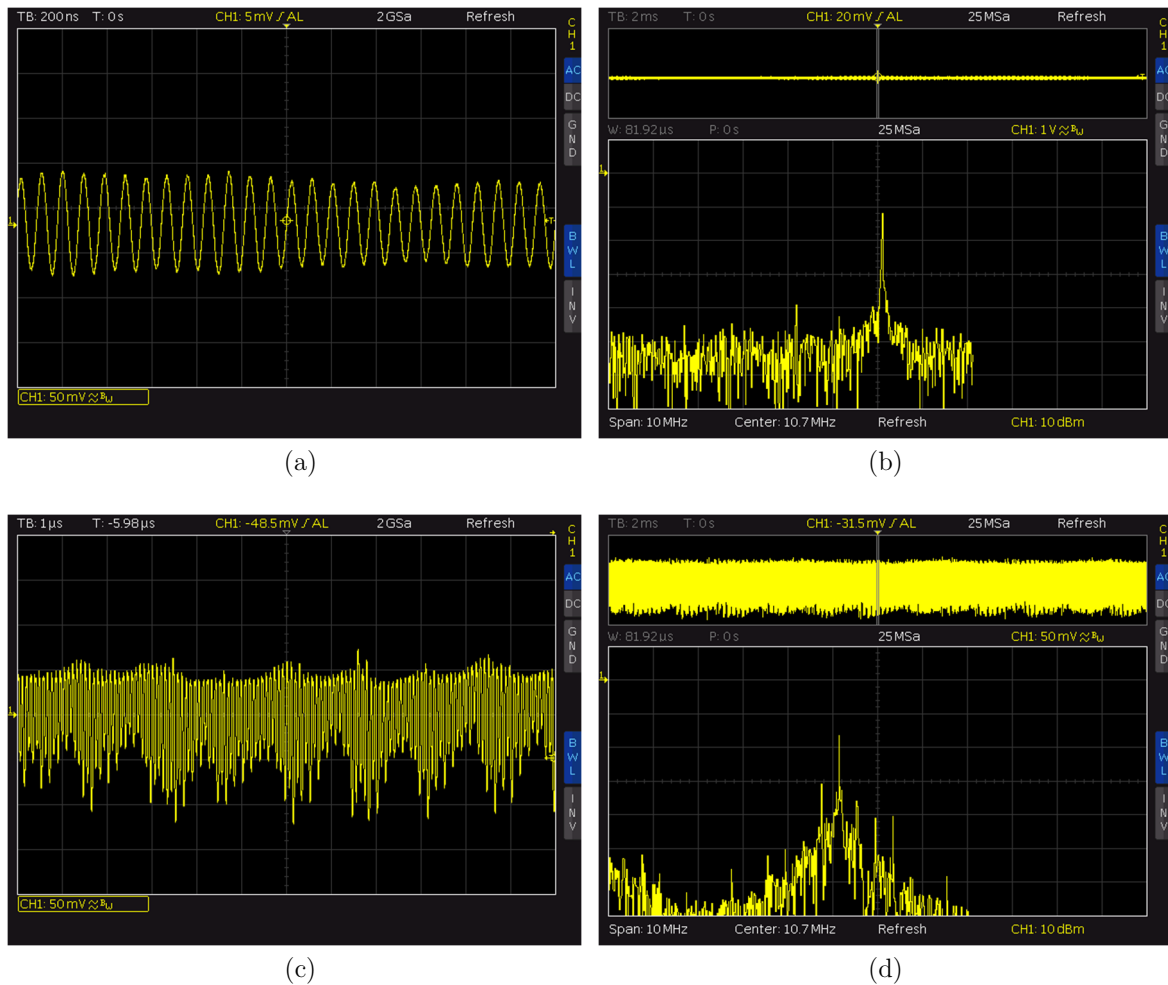


Figura 5.9. (a) Sinal à entrada do amplificador FI. (b) Espectro do sinal à entrada do amplificador FI. (c) Sinal à saída do amplificador FI. (d) Espectro do sinal à saída do amplificador FI.

A figura 5.10 apresenta o sinal IF BFSK à entrada e à saída do limitador e o seu respectivo espectro. É possível observar que o sinal à entrada do limitador já apresenta novamente um aspecto mais desejado, devido à acção proveniente do filtro passa-banda a jusante do amplificador FI. À saída do limitador verifica-se que o sinal é amplificado significativamente e que o mesmo se encontra limitado em amplitude para cerca de 115 mVpp.

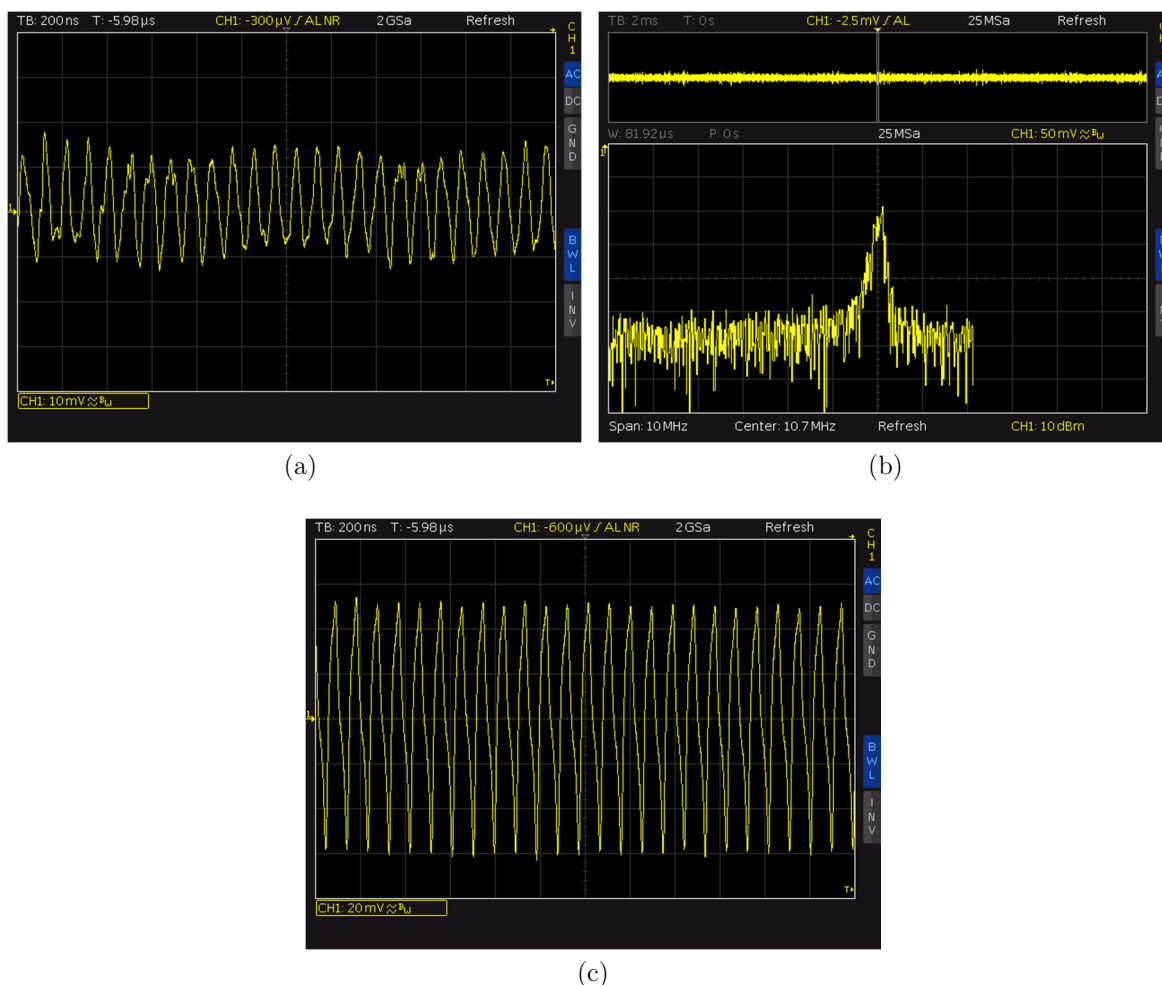


Figura 5.10. (a) Sinal à entrada do limitador. (b) Espectro do sinal à entrada do limitador. (c) Sinal à saída do limitador.

A figura 5.11 apresenta o sinal desmodulado à saída do desmodulador FSK e o sinal desmodulado amplificado à saída do amplificador de banda-base do receptor. Como se pode observar, o sinal desmodulado apresenta uma forma quadrada na qual já é possível distinguir facilmente o sinal original. À saída do amplificador de banda-base verifica-se que o sinal desmodulado é amplificado para cerca do dobro da sua amplitude inicial. Verificou-se na prática que também a componente DC do sinal é amplificada para aproximadamente o dobro mas que, ainda assim, a mesma se encontra abaixo do limiar de decisão do comparador do andar seguinte, como desejado.

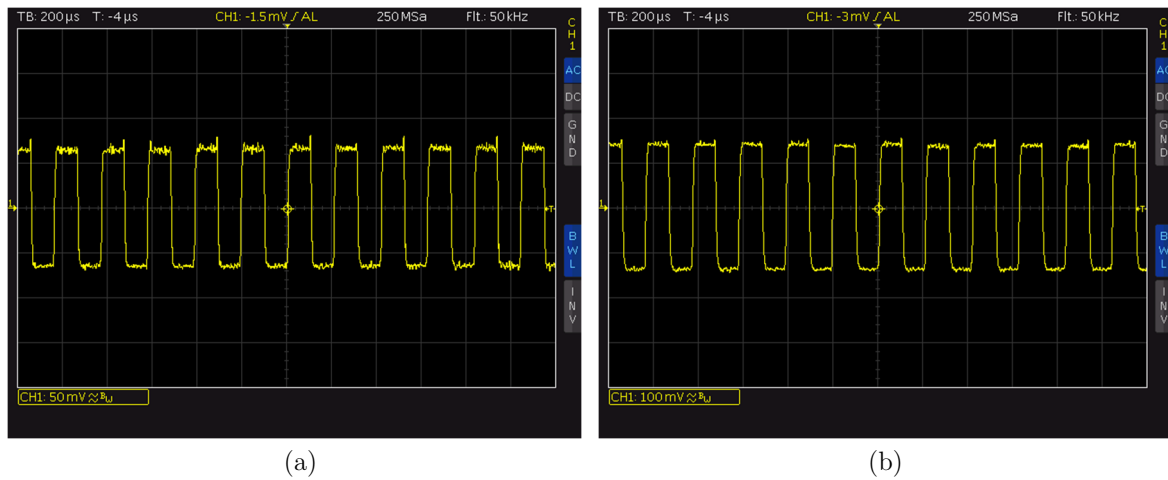


Figura 5.11. (a) Sinal desmodulado à saída do desmodulador FSK. (b) Sinal desmodulado amplificado à saída do amplificador de banda-base do receptor.

A figura 5.12 apresenta o sinal digital binário (trama UART a receber) à saída do comparador, isto é, à saída do circuito de decisão do receptor. Como se pode ver, o sinal é regenerado para níveis superiores e inferiores bem definidos, sendo depois recebido pelo módulo UART do microcontrolador do receptor como um sinal digital binário. O sinal recebido é portanto uma réplica exacta do sinal original enviado pelo emissor.

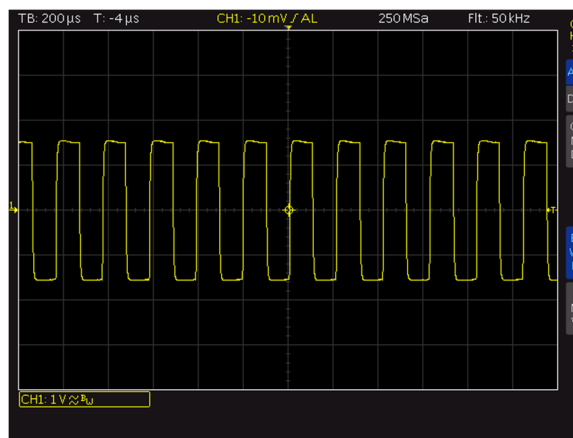


Figura 5.12. Sinal digital binário à saída do comparador.

### 5.2.4 Alcance do transceiver

Para determinar o alcance do *transceiver* foram realizados testes semelhantes para seis cenários diferentes, os quais estão representados na figura 5.13. Em todos os testes foi utilizada novamente uma sequência binária genérica com *baudrate* de 30 Kbps.

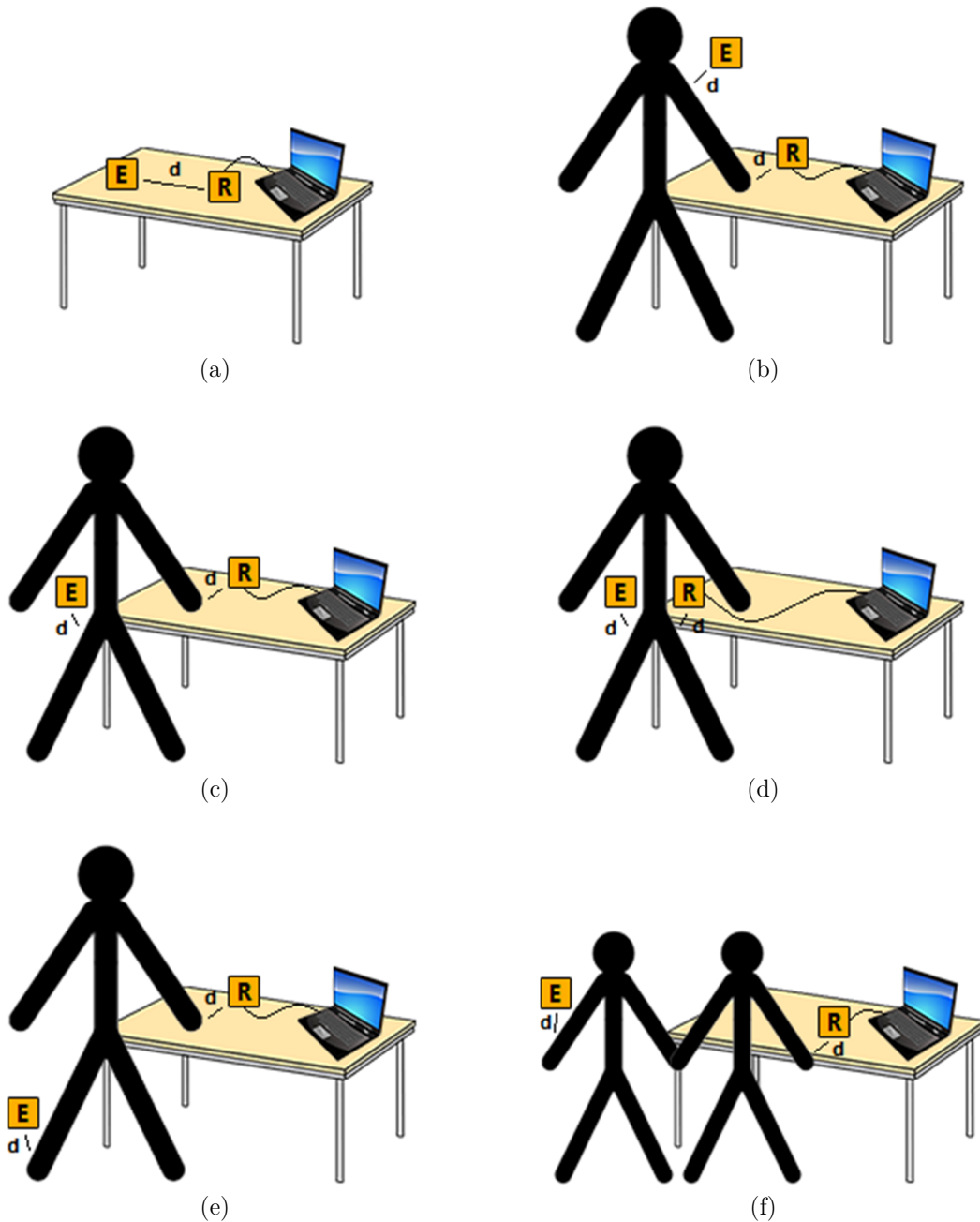


Figura 5.13. Testes efectuados. (a) Teste 1. (b) Teste 2. (c) Teste 3. (d) Teste 4. (e) Teste 5. (f) Teste 6.

O primeiro teste consistiu na colocação de ambos os *transceivers* emissor e receptor sobre uma mesa, onde posteriormente se mediu o nível da RSSI no receptor para diferentes distâncias entre os dois dispositivos. O gráfico da figura 5.14 apresenta a variação do nível da RSSI para as distâncias ( $d$ ) em que não ocorreram erros na transmissão. A verificação dos erros foi efectuada por observação do sinal recebido utilizando um PC ao qual o receptor estava ligado por USB e um osciloscópio. Apesar de empírica, esta abordagem permitiu de facto essa verificação através da visualização da variação dos níveis do sinal em cada amostragem. Convém ainda referir que a mesa utilizada continha um suporte metálico que poderá ter tido alguma influência nos resultados obtidos devido ao acoplamento capacitivo existente.

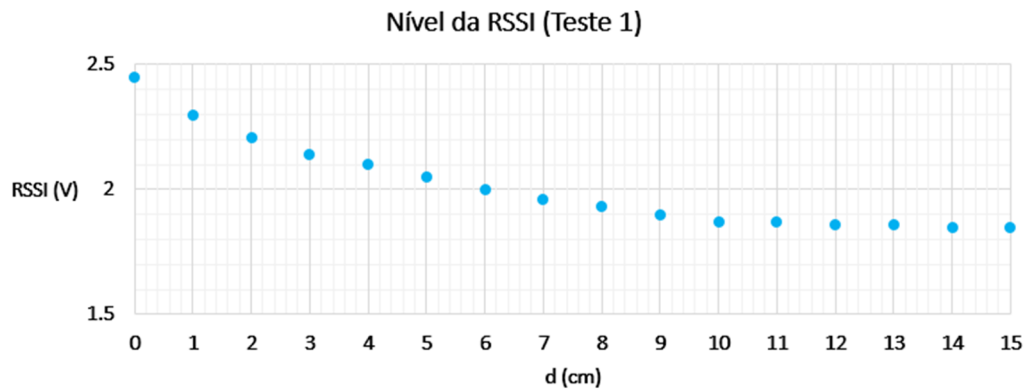


Figura 5.14. Nível da RSSI para diferentes distâncias entre o emissor e o receptor para o caso em que o canal de transmissão é uma superfície não condutora (Teste 1).

Através da figura 5.14 pode afirmar-se que, no caso em que estão sobre uma superfície não condutora, os *transceivers* possuem um alcance de cerca de 15 cm e que o nível da RSSI decresce com a distância de forma aproximadamente linear, situando-se entre 1.8V e 2.5V.

Os testes seguintes consistiram na colocação dos *transceivers* emissor e receptor em diferentes zonas do corpo humano. Mediu-se novamente o nível da RSSI mas desta vez para a distância ( $d$ ) entre os *transceivers* e o corpo humano onde não ocorreram erros na transmissão. A verificação dos erros foi efectuada de forma semelhante ao primeiro teste. Na tabela 5.1, apresentam-se os resultados obtidos para estes testes.

Teste	Distância (d)	Nível da RSSI
Teste 2	< 2 cm	2.10 V
Teste 3	< 2 cm	2.06 V
Teste 4	< 2 cm	2.20 V
Teste 5	< 2 cm	2.08 V
Teste 6	< 2 cm	1.75 V

Tabela 5.1. Nível da RSSI para os casos em que o canal de transmissão é o corpo humano (Testes 2, 3, 4, 5 e 6).

Pela tabela 5.1 conclui-se que, quando acoplados ao corpo humano, os *transceivers* possuem um alcance ao corpo de cerca de 2 cm e que o nível da RSSI é nitidamente menor no caso em que o canal é constituído por dois corpos.

### 5.2.5 Consumo do transceiver

Na tabela 5.2 apresentam-se os resultados relativos ao consumo dos modos emissor e receptor do *transceiver*. Estes resultados foram obtidos utilizando a novamente uma sequência binária genérica com *baudrate* de 30 Kbps e colocando uma resistência de  $1\ \Omega$  entre o terminal positivo da pilha e o pino de alimentação do *transceiver*. Esta abordagem permite medir directamente o consumo de corrente do *transceiver*, pois o valor da corrente que atravessa a resistência é equivalente ao valor da tensão aos seus terminais.

Modo	Consumo
Emissor	19 mA
Receptor	15 mA

Tabela 5.2. Consumo dos modos emissor e receptor do *transceiver*.

Os blocos que mais contribuem para o consumo do *transceiver* no modo emissor são:

- Regulador de tensão (Texas Instruments TLV70030)
- Microcontrolador (Microchip PIC24FJ128GA306 - Módulos principais: UART, FSK Output Compare, Input Change Notification, 1 Timer)
- Multiplicador de tensão (Texas Instruments REG710)
- IF *up-conversion mixer* (NXP SA602A)
- RF *up-conversion mixer* (NXP SA602A)
- Amplificador sintonizado (NXP BFR520)

Por sua vez, os blocos que mais contribuem para o consumo no modo receptor são:

- Regulador de tensão (Texas Instruments TLV70030)
- Microcontrolador (Microchip PIC24FJ128GA306 - Módulos principais: UART, Varicap Tuning Output Compare, 2 Comparators, ADC, 1 Timer)
- Amplificador sintonizado (NXP BF998)
- Andar de FI (NXP SA636)
- Amplificador de banda-base (Texas Instruments LMH6647)

## 5.3 Sistema de pagamento

### 5.3.1 Modelação inicial (sem transceivers)

Antes de se realizarem as experiências para o sistema de pagamento final com os *transceivers* integrados na sua arquitectura, primeiro é necessário efectuar os testes que confirmem que o mesmo funciona sem os *transceivers*. Como tal, modelou-se o sistema de pagamento sem *transceivers* através dos seguintes testes:

- Teste em *loopback*
- Teste com ligação *shunt*

O teste em *loopback* consiste em utilizar apenas um *smartphone* e uma placa IOIO-OTG onde a ligação entre o pino transmissor e o pino receptor da placa é feita directamente através de um fio condutor. Este teste teve como principal objectivo testar a comunicação entre um *smartphone* e uma placa IOIO-OTG utilizando tramas de dados UART. Nesse sentido, desenvolveu-se uma aplicação Android para testar a comunicação onde se configurou apenas um módulo UART da placa IOIO-OTG. A figura 5.15 mostra a configuração utilizada para o teste em *loopback*. Neste teste, o *smartphone* utilizado foi um Samsung Galaxy S3.

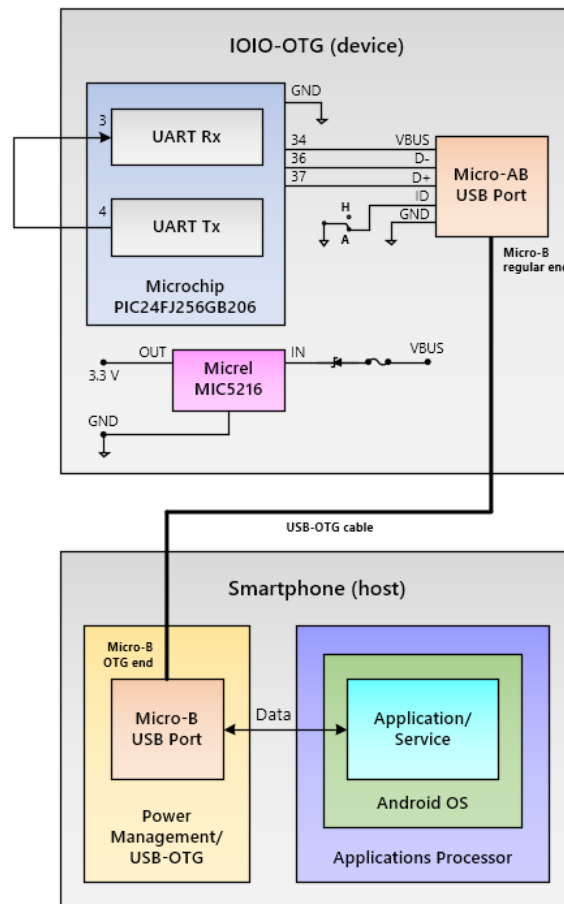


Figura 5.15. Configuração utilizada para o teste em *loopback*.

A aplicação desenvolvida permite trocar tramas de dados UART entre dois dispositivos Android utilizando placas IOIO-OTG. No entanto, como neste caso é utilizado apenas um *smartphone*, então este será tanto o dispositivo emissor como o dispositivo receptor. A trama a enviar/receber pelo mesmo será a que se apresenta na figura 5.16.

Repare-se que a trama configurada segue a norma definida para as tramas do sistema de pagamento implementado. É possível observar os delimitadores e os campos do nome, número de conta, saldo e *password* (ou PIN). Refere-se ainda que a secção do código utilizado para



transmitir e receber as tramas é semelhante ao que foi desenvolvido para as aplicações BCC Terminal e BCC Wallet.

Ao ser clicado o botão que activa o modo de transmissão/recepção do dispositivo, será enviada a trama apenas uma vez. Caso a comunicação seja bem-sucedida, os campos da trama surgirão separadamente no ecrã do *smartphone* a par da informação de que a operação foi concluída. Como se pode ver pela figura 5.17, os resultados mostram que o teste foi bem-sucedido.

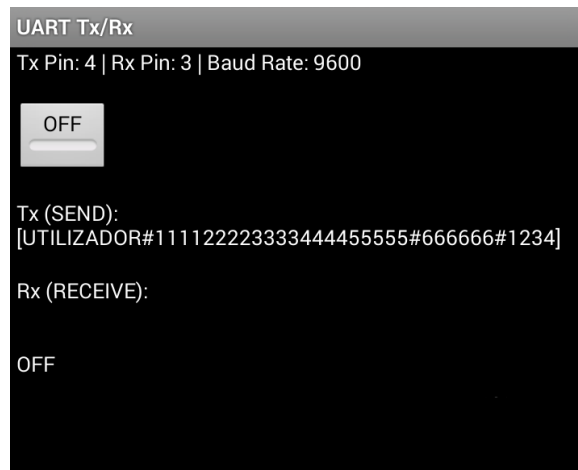


Figura 5.16. Trama enviada (teste em *loopback*).

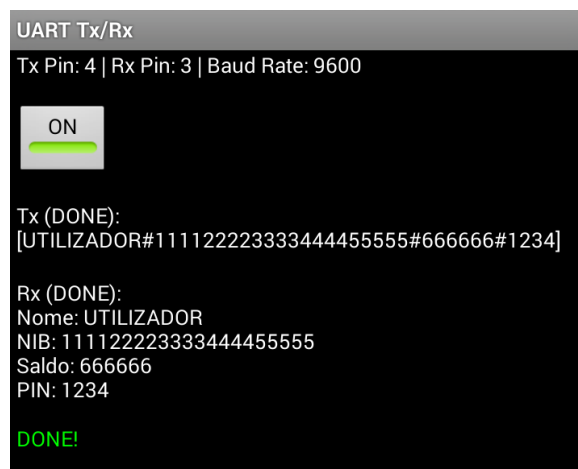


Figura 5.17. Trama recebida (teste em *loopback*).

Em breve nota, menciona-se que foram também testadas as aplicações BCC Terminal e BCC Wallet utilizando esta configuração. Isto apenas foi possível simulando via *software* o utilizador (cliente) no código da aplicação BCC Terminal e a entidade vendedora no código da aplicação BCC Wallet. Os resultados obtidos foram os pretendidos mas, como também foi efectuado o teste com ligação *shunt*, apenas se irá apresentar os resultados para este último pois foram obtidos no mesmo contexto.

O teste com ligação *shunt* consiste por sua vez em utilizar dois *smartphones* e duas placas IOIO-OTG onde o pino transmissor de cada placa é ligado directamente ao pino receptor da

outra placa através de um fio condutor. Esta configuração, apresentada na figura 5.18, já se aproxima mais do modelo final do sistema de pagamento implementado, apenas com a diferença de que não são utilizados *transceivers* e o canal de transmissão é simplesmente um par de fios condutores ao invés do corpo humano.

Realça-se ainda que este teste é de extrema importância pois vai permitir determinar se o sistema de pagamento foi correctamente implementado antes de se integrarem os *transceivers* na sua arquitectura.

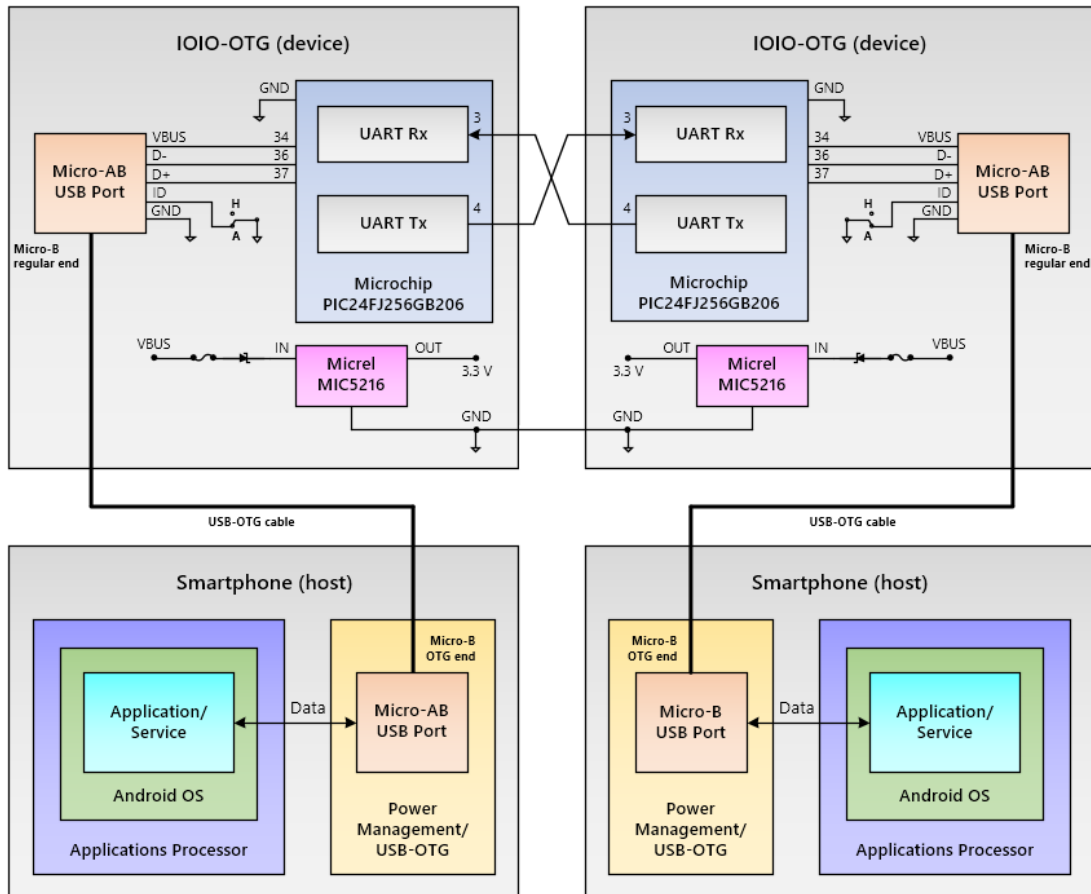


Figura 5.18. Configuração utilizada para o teste com ligação *shunt*.

Os procedimentos para este teste foram os mesmos que se realizaram para o teste em *loopback*. Neste caso, cada placa IOIO-OTG tem o seu módulo UART configurado e ambos os *smartphones* correm a aplicação de teste desenvolvida. Os *smartphones* utilizados foram dois Samsung Galaxy S3. De forma equivalente às figuras 5.16 e 5.17, os resultados mostram que também este teste foi bem-sucedido.

Tal como para o caso anterior, também foram testadas as aplicações BCC Terminal e BCC Wallet mas agora utilizando a configuração do teste com ligação *shunt*. Como neste caso são utilizados dois *smartphones*, então já não é necessário efectuar qualquer simulação, pelo que um dos *smartphones* será o terminal de pagamentos da entidade vendedora e o outro a carteira digital do utilizador (cliente). Assim, o primeiro correrá a aplicação BCC Terminal e o segundo a aplicação BCC Wallet.

O processo de pagamento conduzido neste teste é o mesmo que se encontra descrito no capítulo 4. Além disso, foram também testadas todas as operações que poderiam originar erros. A tabela 5.3 apresenta os resultados obtidos para as aplicações BCC Terminal e BCC Wallet. Como se pode verificar, os resultados mostram que o sistema de pagamento implementado funciona na perfeição independentemente da integração dos *transceivers* no seu modelo final.

**BCC Terminal - MainActivity**

Operação	Resultado	Sucesso
MainActivity onStart()/onResume()	Quantia inserida nula	Sim
Clicar botão “Efectuar pagamento” com quantia inserida nula	Apresenta mensagem: “Inserir quantia”	Sim
Clicar botão “Efectuar pagamento” com quantia inserida não nula	Abre a ConfirmActivity	Sim
Clicar botão “C”	Apaga a quantia inserida	Sim
Clicar botão “.”	Apresenta mensagem: “Função não implementada”	Sim
Clicar botão “Definições”	Abre a SettingsActivity	Sim

**BCC Terminal - ConfirmActivity**

Operação	Resultado	Sucesso
Comunicação correcta entre os dispositivos	Apresenta a informação correspondente e o botão “Confirmar pagamento”	Sim
Comunicação incorrecta ou inexistente entre os dispositivos	Abre a MainActivity e apresenta mensagem: “Utilizador não detectado”	Sim
Temporizador atinge o tempo pré-definido	Cancela todas as operações, abre a MainActivity e apresenta mensagem: “Utilizador não detectado”	Sim
Clicar botão “Confirmar”	Abre a EnterPinActivity	Sim
Clicar botão “Cancelar”	Abre a MainActivity	Sim

**BCC Terminal - EnterPinActivity**

Operação	Resultado	Sucesso
Inserir PIN incompleto ou não inserir PIN e clicar botão “OK”	Apresenta mensagem: “Inserir PIN”	Sim
Inserir PIN completo e clicar botão “OK”	Abre a PayActivity	Sim
Clicar botão “C”	Apaga os dígitos inseridos	Sim
Clicar botão “Cancelar”	Cancela todas as operações e abre a MainActivity	Sim

### BCC Terminal - PayActivity

Operação	Resultado	Sucesso
Comunicação correcta entre os dispositivos	Incrementa o saldo da entidade vendedora com a quantia inserida e apresenta a factura e o botão “OK”	Sim
Comunicação incorrecta ou inexistente entre os dispositivos	Abre a MainActivity e apresenta mensagem: “Operação anulada”	Sim
Recebido comando para abortar o processo de pagamento após a introdução de um PIN incorrecto	Abre a MainActivity e apresenta mensagem: “PIN incorrecto”	Sim
Temporizador atinge o tempo pré-definido	Cancela todas as operações, abre a MainActivity e apresenta mensagem: “Operação anulada”	Sim
Clicar botão “OK”	Abre a MainActivity	Sim
Clicar botão “Cancelar”	Cancela todas as operações e abre a MainActivity	Sim

### BCC Wallet - MainActivity

Operação	Resultado	Sucesso
MainActivity onStart()/onResume()	Apresenta o saldo actualizado do utilizador e a lista actualizada dos pagamentos efectuados	Sim
Clicar botão “Activar/Desactivar” e inserir PIN correcto	Activa/Desactiva o AppService	Sim
Clicar botão “Activar/Desactivar” e inserir PIN incorrecto	Apresenta mensagem: “Password incorrecta”	Sim
Clicar botão “Apagar lista” e confirmar via botão “Sim”	Apaga todo o conteúdo da base de dados e apresenta uma lista vazia	Sim
Clicar botão “Apagar lista” e cancelar via botão “Não”	Cancela a operação	Sim
Clicar botão “Definições”	Abre a SettingsActivity	Sim
Clicar linha da lista	Abre a PayActivity	Sim

### BCC Wallet - AppService

Operação	Resultado	Sucesso
Comunicação correcta entre os dispositivos	Decrementa o saldo do utilizador com a quantia inserida, armazena a factura na base de dados, apresenta a notificação de pagamento efectuado e bloqueia os pagamentos por um tempo pré-definido	Sim

Comunicação incorrecta ou inexistente entre os dispositivos	Cancela todas as operações e/ou aguarda por nova comunicação	Sim
PIN recebido incorrecto	Envia comando para abortar o processo de pagamento, cancela todas as operações e aguarda por nova comunicação	Sim
Temporizador atinge o tempo pré-definido	Desbloqueia os pagamentos	Sim

BCC Wallet - PayActivity

Operação	Resultado	Sucesso
Linha da lista da MainActivity clicada	Apresenta a factura correspondente e o botão “OK”	Sim
Notificação proveniente do AppService clicada	Apresenta a factura correspondente e o botão “OK”	Sim
Clicar botão “OK”	Abre a MainActivity	Sim

Tabela 5.3. Resultados obtidos para o teste com ligação *shunt*.

A tabela 5.4 resume e compara os resultados obtidos nos dois testes efectuados.

Teste	Efectua comunicações sem erros (Experiência 1)	Efectua pagamentos sem erros (Experiência 2)
<i>Loopback</i>	Sim	Sim
Ligação <i>shunt</i>	Sim	Sim

Tabela 5.4. Comparação entre os resultados obtidos para os dois testes efectuados.

### 5.3.2 Sistema final (com transceivers)

Nesta secção apresentam-se as experiências a realizar com o modelo final do sistema de pagamento, isto é, com o sistema de pagamento com os *transceivers* integrados. Este tem por base a arquitectura e a interface de interligação apresentadas nas figuras 4.5 e 4.9, respectivamente, pelo que sua a configuração final será então a que se apresenta na figura 5.19.

Para testar o sistema realizam-se as mesmas experiências que foram conduzidas nos testes da secção anterior. Portanto, na primeira experiência é utilizada a aplicação Android desenvolvida para testar a comunicação entre os dispositivos e na segunda experiência é testado efectivamente o modelo final, onde um dos dispositivos corre a aplicação BCC Terminal e o outro a aplicação BCC Wallet. Em ambas as experiências, os *transceivers* encontram-se configurados simultaneamente como emissor e receptor e estão acoplados ao corpo humano a uma distância deste menor que 2 cm, de forma a evitar a ocorrência de erros. Por sua vez, os *smart-phones* encontram-se configurados como *host*.

Os resultados a obter deveriam ser semelhantes aos dos testes *loopback* e ligação *shunt* (apresentados nas figuras 5.16 e 5.17 e nas tabelas 5.3 e 5.4). Contudo, por razões que transcendem a realização este projecto, não foi possível efectuar os testes mencionados acima. Deste modo, apenas se refere que os mesmos deverão ser realizados como trabalho futuro.

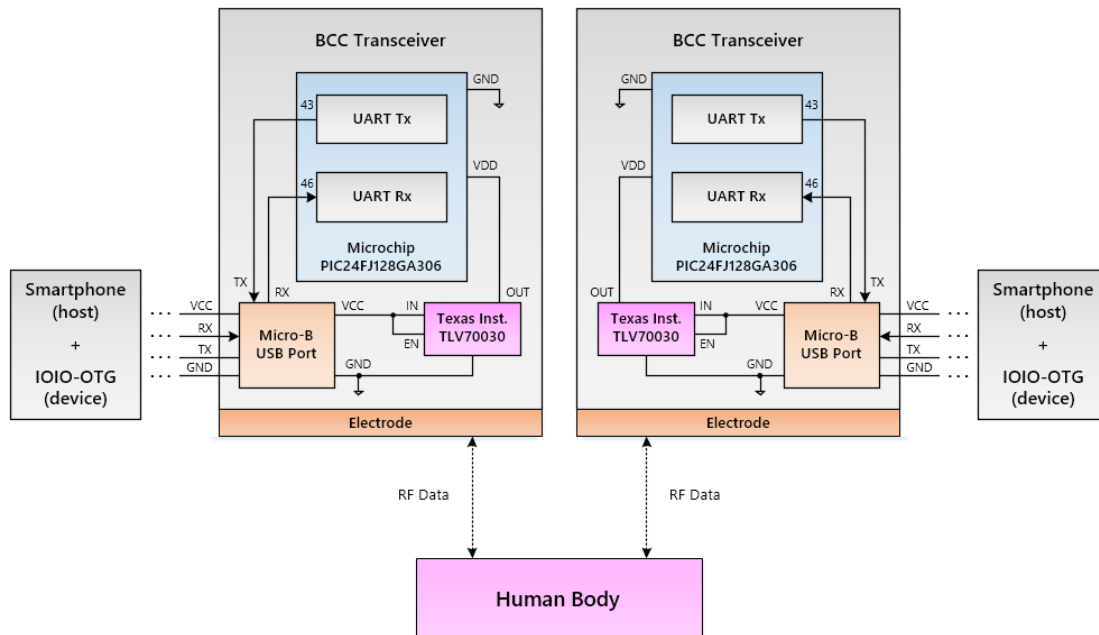


Figura 5.19. Configuração final do sistema de pagamento implementado.

## 5.4 Considerações finais

Perante os resultados obtidos neste capítulo, pode afirmar-se que a implementação dos *transceivers* do sistema BCC foi bem-sucedida. Para o *baudrate* testado (30 Kbps), estes tornaram possível a transmissão e a recepção de sinais sem erros por acoplamento capacitivo ao corpo humano para um alcance ao corpo máximo de cerca de 2 cm. É de realçar que as melhorias efectuadas ao *transceiver*, entre elas, a substituição do microcontrolador, a substituição dos cristais e a redução do tamanho da PCB revelaram-se uma mais-valia, pois o *transceiver* actual consome menos potência que os das suas versões anteriores.

Por sua vez, também foram obtidos bons resultados para o sistema de pagamento implementado. Este, numa primeira abordagem sem *transceivers*, permitiu a troca de informação e a realização de pagamentos entre os dispositivos Android. Depois, na sua versão final já com os *transceivers* integrados, deveria permitir novamente a troca de informação entre os dispositivos e a realização de pagamentos através do corpo humano quando utilizada a configuração necessária para a correcta comunicação entre os *transceivers*. Contudo, não foi possível testar o sistema de pagamento quando interligado ao sistema BCC, pelo que este teste deverá ser efectuado no futuro.

No caso do sistema de pagamento, dada a enorme dificuldade e a exagerada extensão que a apresentação dos resultados sob a forma de figuras viria a trazer para este capítulo, optou-se simplesmente por mencionar se os testes efectuados foram ou não bem-sucedidos e por apresentar os resultados sob a forma de tabelas contendo uma confirmação positiva ou negativa de que os principais objectivos foram atingidos. De igual modo, sempre que surgisse um caso em que iria ser replicada a apresentação de resultados anteriores, optou-se simplesmente por fazer referência às figuras e tabelas que já continham esses mesmos resultados.

Apesar de não fornecer uma visualização tão intuitiva, esta abordagem permitiu a apresentação dos resultados de forma concisa e compacta.

## Capítulo 6

# Conclusão e trabalho futuro

### 6.1 Sumário

O presente capítulo é o que encerra esta dissertação. É feito um resumo geral de todo o trabalho desenvolvido e são elaborados os últimos comentários relativamente aos sistemas de comunicação e pagamento implementados. No final são identificados os aspectos a melhorar nos dois sistemas e são propostas algumas possíveis implementações futuras que poderão dar continuidade a este projecto.

### 6.2 Conclusão

Esta dissertação de mestrado teve como principal objectivo o desenvolvimento de um sistema de pagamento seguro utilizando comunicação através do corpo humano. Como objectivo secundário teve o desenvolvimento de uma nova versão do *transceiver* do sistema de comunicação por acoplamento capacitivo da Universidade de Aveiro, no sentido de se reduzir o seu tamanho e consumo de energia.

Em primeiro lugar, realizou-se um estudo ao estado da arte das tecnologias BCC e dos sistemas de pagamento móveis. Depois efectuou-se uma análise detalhada ao sistema BCC da Universidade de Aveiro, procedendo-se de seguida à sua melhoria e ao desenvolvimento de novas implementações. Feito isto, deu-se início ao desenvolvimento do sistema de pagamento onde se começou por explicar o seu conceito, a sua arquitectura e os processos envolvidos no seu funcionamento. Recorrendo a placas IOIO-OTG, implementou-se a interface Android do sistema de pagamento, a qual interliga os *transceivers* do sistema BCC aos dispositivos Android das entidades envolvidas no processo de pagamento. Daqui resultou o desenvolvimento das aplicações BCC Terminal e BCC Wallet, as quais processam todas as fases do pagamento e permitem trocar a informação entre os dispositivos.

Finalizadas todas as implementações relativas aos dois sistemas, procedeu-se à realização dos testes que permitiram retirar conclusões acerca do funcionamento de ambos. Em relação ao sistema BCC, para o *baudrate* testado (30 Kbps) os resultados obtidos confirmam que o mesmo é capaz de transmitir e receber informação sem erros por acoplamento capacitivo ao corpo humano até um alcance ao corpo máximo de cerca de 2 cm. Conclui-se também que quantos mais corpos constituírem o canal de transmissão, menor será o nível de potência do sinal recebido. Relativamente ao sistema de pagamento, os resultados obtidos também de-

monstram que este funciona correctamente, ou seja, o sistema implementado permite trocar informação e efectuar pagamentos entre dois dispositivos Android com um certo grau de segurança. Contudo, faltou testar a interligação entre os dois sistemas implementados, algo que deverá ser realizado como trabalho futuro.

Por fim, é importante mencionar que os dois sistemas implementados no âmbito desta dissertação de mestrado ainda possuem grandes margens para melhorias e novas implementações, pois ainda se encontram nas suas versões protótipo. No entanto, ambos representam soluções promissoras para um futuro próximo e um objecto de estudo que pode contribuir fortemente para a investigação e desenvolvimento nas áreas dos sistemas de pagamento móveis, Android e BCC.

## 6.3 Trabalho futuro

### 6.3.1 Transceiver low-power

O baixo consumo foi sempre um dos principais objectivos a alcançar na implementação do *transceiver*. Nesse sentido, desenvolveu-se uma segunda versão do *transceiver*, a que se designou de *low-power*, onde se efectuaram algumas modificações relativamente à primeira versão (a que se designou de *standard*). Contudo, não foi possível concluir a sua implementação, pelo que aqui apenas se deixa a sua descrição resumida, a qual poderá servir de base para trabalhos futuros.

O *transceiver* da versão *low-power* poderá ser idêntico ao da versão *standard*, apenas com a diferença de que os andares de IF e RF *up-conversion* poderão ser implementados de outra forma. Ao invés do *mixers* NXP SA602A, nesta segunda versão poderiam ser utilizados circuitos analógicos de consumo reduzido especialmente desenhados para efectuar as conversões de frequência para uma menor tensão de alimentação. Assim, a arquitectura apresentada na figura 3.2 manteria-se para esta versão, apenas com a excepção de que já não existiria a correspondência do NXP SA602A aos andares de *up-conversion*, pois o mesmo seria removido nesta versão. Quanto à alimentação do *transceiver*, as implementações da versão *standard* também se aplicariam à versão *low-power*, apenas com a excepção de que nesta versão seria removido o multiplicador de tensão (*charge pump*), pois os andares de IF e RF *up-conversion* já não necessitariam de ser alimentados a 5V. De igual modo, o amplificador sintonizado do emissor deixaria de ser alimentado a 5V, passando a estar alimentado a 3V. Consequentemente, a par da redução do consumo do *transceiver* também poderia ser reduzido o tamanho da PCB, após a remoção dos componentes mencionados.

### 6.3.2 Sistema de pagamento

Futuramente, novas funcionalidades poderão ser introduzidas no sistema de pagamento implementado. Entre elas, menciona-se em primeiro lugar a criação de uma base de dados dos pagamentos efectuados para a aplicação BCC Terminal e a possibilidade de identificar o utilizador através da sua impressão digital utilizando um *smartphone* com módulo para reconhecimento de impressões digitais e desenvolvendo posteriormente o *software* para o efeito. Outra sugestão seria tentar encontrar-se uma solução que permitisse englobar num dispositivo único (*smartphone* no caso ideal) todo o *hardware* necessário para o funcionamento do sistema de pagamento. Uma implementação de relevo seria também a reformulação da janela das defini-



ções e do seu respectivo *software* em ambas as aplicações BCC Terminal e BCC Wallet no sentido de eliminar a funcionalidade de teste actual e de passar a registar/autenticar a entidade vendedora e o utilizador (cliente) através de uma comunicação *web* com uma entidade bancária. A partir dessa comunicação, os dados e o saldo da entidade ou do utilizador seriam actualizados continuamente tanto nos servidores da entidade bancária como nos *smartphones* dos primeiros.

Seria também importante a implementação de um algoritmo de encriptação para a comunicação, algo que permitiria aumentar bastante a segurança do sistema. Sugere-se ainda a implementação de uma PCB única para a interface de interligação do sistema de pagamento contendo todos os componentes do *transceiver* desenvolvido e da placa IOIO-OTG. Para finalizar, relembra-se que deverão ser efectuados os testes à interface de interligação desenvolvida para o sistema de pagamento, os quais poderão confirmar se o mesmo permite a realização de pagamentos através do corpo humano quando interligado ao sistema BCC.



# Bibliografia

- [1] J. H. Donker, “The body as a communication medium”, 11th Twente Student Conference on IT, 2009.
- [2] Vimeo - IBM: Tom Zimmerman. Online:  
[http://i.vimeocdn.com/video/241932184\\_640.jpg](http://i.vimeocdn.com/video/241932184_640.jpg)
- [3] T. G. Zimmerman, “Personal Area Networks (PAN): Near-field Intra-Body Communication”, Master’s thesis, Massachusetts Institute of Technology, 1995.
- [4] T. G. Zimmerman, “Personal Area Networks: Near-field intrabody communication”, IBM Systems Journal, vol. 35, 1996.
- [5] M. R. Yuce, J. Y. Khan, “Wireless Body Area Networks: Technology, Implementation, and Applications”, Pan Stanford Publishing, 2011.
- [6] Renesas - Renesas Solutions for Wireless Sensor Networks - Part 2: Body Area Networks. Online: [http://tw.renesas.com/edge\\_ol/features/08/index.jsp](http://tw.renesas.com/edge_ol/features/08/index.jsp).
- [7] M. Hanson, H. Powell, A. Barth, K. Ringgenberg, B. Calhoun, J. Aylor, J. Lach, “Body Area Sensor Networks: Challenges and Opportunities”, IEEE Computer Society, 2009.
- [8] V. Custodio, F. Herrera, G. López J. Moreno, “A Review on Architectures and Communications Technologies for Wearable Health-Monitoring Systems”, Carlos III University of Madrid, 2012.
- [9] S. Movassaghi, M. Abolhasan, J. Lipman, D. Smith, A. Jamalipour, “Wireless Body Area Networks: A Survey”, IEEE Communications Surveys & Tutorials, 2013.
- [10] B. Latré, B. Braem, I. Moerman, C. Blondia, P. Demeester, “A Survey on Wireless Body Area Networks”, Wireless Networks, vol. 17, 2010.
- [11] Z. W. Bhatti, “Radio-Frequency and Body-Coupled Communication Dual Technology Wireless Sensor System Architecture”, Master’s thesis, Royal Institute of Technology, 2009.
- [12] M. H. Maruf, “An Input Amplifier for Body-Channel Communication”, Master’s thesis, Linköping University, 2013.
- [13] E. Karulf, “Body Area Networks (BAN)”, 2008.
- [14] Y. Kado, M. Shinagawa, “RedTacton Near-body Electric-field Communications Technology and Its Applications”, NTT Technical Review, vol. 8, no. 3, 2010.
- [15] Renesas - Renesas Solutions for Wireless Sensor Networks - Part 1: Overview. Online: [http://tw.renesas.com/edge\\_ol/features/07/index.jsp](http://tw.renesas.com/edge_ol/features/07/index.jsp)

- [16] N. S. Mazloun, "Body-Coupled Communications: Experimental characterization, channel modeling and physical layer design", Master's thesis, Chalmers University of Technology, 2009.
- [17] S. Corroy, A. Argyriou, Z. Wadood, H. Baldus, "A Body-Coupled Communication and Radio Frequency Dual Technology Cooperation Protocol for Body-Area Networks", Second International Workshop on Medical Applications Networking, 2010.
- [18] Y. Lee, J. Park, G. Yoon, "Biosignal Measurement using Personal Area Network based on Human Body Communication", International Science Index Journal, vol. 7, 2013.
- [19] M. Wegmüller, W. Fichtner, P. Niederer, N. Kuster, "Intra-Body Communication for Biomedical Sensor Networks", ETH Zurich, 2007.
- [20] M. Wegmüller, A. Kuhn, J. Froehlich, M. Oberle, N. Felber, N. Kuster, W. Fichter, "An Attempt to Model the Human Body as a Communication Channel", IEEE Transactions on Biomedical Engineering, vol. 54, 2007.
- [21] H. Baldus, S. Corroy, "Human-centric connectivity enabled by body-coupled communications", Communications Magazine, IEEE, vol. 47, 2009.
- [22] T. Tilak, K. Rahate, P. Sawant, "Comparison of approaches to intra-body communication", Indian Journal of Computer Science and Engineering, vol. 3, no. 5, 2012.
- [23] A. Umenei, "Understanding Low Frequency Non-Radiative Power Transfer", Wireless Power Consortium, Fulton Innovation, 2011.
- [24] C. Capps, "Near field or far field?", 2001.
- [25] K. Ito, N. Haga, M. Takahashi, K. Saito, "Evaluations of Body-Centric Wireless Communication Channels in a Range From 3 MHz to 3 GHz", Proceedings of the IEEE, vol. 100, no. 7, 2012.
- [26] N. Cho, J. Yoo, S. Song, J. Lee, S. Jeon, H. Yoo, "The Human Body Characteristics as a Signal Transmission Medium for Intrabody Communication", IEEE transactions on microwave theory and techniques, vol. 55, no. 5, 2007.
- [27] S. Jeon, C. Yu, Y. Suh, J. Moon, S. Lee, "Analysis of Body Communication Parameters Using Software Radio Platform", Military Communications Conference 2011, 2011.
- [28] K. Hachisuka, A. Nakata, T. Takeda, Y. Terauchi, K. Shiba, K. Sasaki, H. Hosaka, K. Itao, "Development and performance analysis of an intra-body communication device", The 12th International Conference on Solid State Sensors, Actuators and Microsystems, 2003.
- [29] J. Ruiz, J. Xu, S. Shimamoto, "Propagation Characteristics of Intra-body Communications for Body Area Networks", Waseda University, 2006.
- [30] E. Shih, S. Cho, N. Ickes, R. Min, A. Sinha, A. Wang, A. Chandrakasan, "Physical layer driven protocol and algorithm design for energy-efficient wireless sensor networks", Proceedings of the 7th annual international conference on mobile computing and networking, 2001.
- [31] P. Kopecek, "Transmission characteristics of the human body", CSCS '11 Proceedings of the 2nd international conference on Circuits, systems, control, signals, 2011.

- 
- [32] R. Matias, R. Martins, M. Cunha, “Modeling capacitive coupling systems for body coupled communications”, BodyNets ‘12 Proceedings of the 7th International Conference on Body Area Networks, 2012.
- [33] H. A. L. Barreira, “Comunicação através do corpo humano: Desenvolvimento de um transceiver”, Tese de Mestrado, Universidade de Aveiro, 2009.
- [34] J. E. R. Assunção, “Comunicação através do corpo humano: Simulação de um transceiver”, Tese de Mestrado, Universidade de Aveiro, 2010.
- [35] H. F. Castanheira, “Transceiver para comunicação acoplada ao corpo humano”, Tese de Mestrado, Universidade de Aveiro, 2012.
- [36] J. M. L. Lima, “Comunicação acoplada ao corpo humano: Transceiver e Protocolo de rede”, Tese de Mestrado, Universidade de Aveiro, 2013.
- [37] BitScan - How many people really own bitcoins - and why does it matter?. Online: <http://bitscan.com/bitnews/item/how-many-people-really-own-bitcoins-and-why-does-it-matter>
- [38] Moneero - We forget that Bitcoin is One Million Users Young. Online: <https://moneero.com/web/blog/post/we-forget-bitcoin-is-one-million-users-young>
- [39] Wikipedia - Conta corrente. Online: [http://pt.wikipedia.org/wiki/Conta\\_corrente](http://pt.wikipedia.org/wiki/Conta_corrente)
- [40] Wikipedia - Wireless Application Protocol. Online: [http://en.wikipedia.org/wiki/Wireless\\_Application\\_Protocol](http://en.wikipedia.org/wiki/Wireless_Application_Protocol)
- [41] Wikipedia - Near field communication. Online: [http://en.wikipedia.org/wiki/Near\\_field\\_communication](http://en.wikipedia.org/wiki/Near_field_communication)
- [42] MobilePaymentsToday - Visa contactless transactions in the UK nearly quadrupled in 2013. Online: <http://www.mobilepaymentstoday.com/news/visa-contactless-transactions-in-the-uk-nearly-quadrupled-in-2013>
- [43] Visa Europe - Contactless and mobile payments in 2013. Online: <http://www.visaeurope.com/idoc.ashx?docid=ea3de2c9-f3bd-4754-9115-09601c196e9a&version=-1>
- [44] Visa Europe - Visa and Samsung Sign Global Alliance Agreement to Accelerate Mobile (NFC) Payments. Online: [http://www.visaeurope.com/en/newsroom/news/articles/2013/visa\\_and\\_samsung\\_nfc\\_alliance.aspx](http://www.visaeurope.com/en/newsroom/news/articles/2013/visa_and_samsung_nfc_alliance.aspx)
- [45] NFC World - Weve partners with MasterCard for NFC payments in the UK. Online: <http://www.nfcworld.com/2014/02/06/327765/weve-partners-mastercard-nfc-payments-uk/>
- [46] EE Times - How NFC can to speed Bluetooth transactions today. Online: [http://www.eetimes.com/document.asp?doc\\_id=1273409](http://www.eetimes.com/document.asp?doc_id=1273409)
- [47] Sony - NFC Forum Specifications. Online: <http://www.sony.net/Products/felica/NFC/forum.html>

- [48] Oracle - An Introduction to Near-Field Communication and the Contactless Communication API. Online:  
<http://www.oracle.com/technetwork/articles/javame/nfc-140183.html>
- [49] NXP Semiconductors, “SA602A Double-balanced mixer and oscillator”, 2014.
- [50] Murata, “Ceramic Filters (CERAFIL) / Ceramic Discriminators for Communications Equipment”, 2011.
- [51] NXP Semiconductors, “SA636 Low voltage high performance mixer FM IF system with high-speed RSSI”, 2012.
- [52] Microchip, “PIC24FJ128GA310 FAMILY 64/80/100-Pin, General Purpose, 16-Bit Flash Microcontrollers with LCD Controller and nanoWatt XLP Technology”, 2011.
- [53] Texas Instruments, “200-mA, Low-IQ, Low-Dropout Regulator for Portable Devices”, 2012.
- [54] Microchip, “MCP73831/2 Miniature Single-Cell, Fully Integrated Li-Ion, Li-Polymer Charge Management Controllers”, 2013.
- [55] Texas Instruments, “60mA, 5.0V, Buck/Boost Charge Pump in ThinSOT-23 and ThinQFN”, 2009.
- [56] Wikipedia - Mobile operating system. Online:  
[http://upload.wikimedia.org/wikipedia/commons/thumb/8/83/World\\_Wide\\_Smartphone\\_Sales.png/700px-World\\_Wide\\_Smartphone\\_Sales.png](http://upload.wikimedia.org/wikipedia/commons/thumb/8/83/World_Wide_Smartphone_Sales.png/700px-World_Wide_Smartphone_Sales.png)
- [57] Microchip, “PIC24FJ256GB210 FAMILY 64/100-Pin, 16-Bit Flash Microcontrollers with USB On-The-Go (OTG)”, 2010.
- [58] GitHub - IOIO Over OpenAccessory. Online:  
<https://github.com/ytai/ioio/wiki/IOIO-Over-OpenAccessory>
- [59] GitHub - Power Supply (IOIO-OTG). Online:  
<https://github.com/ytai/ioio/wiki/Power-Supply-OTG>
- [60] Sparkfun - IOIO-OTG Schematic. Online:  
<http://dlnmh9ip6v2uc.cloudfront.net/datasheets/Dev/Android/IOIO-OTG-v20b.pdf>

## Apêndice A

# Esquemáticos do transceiver

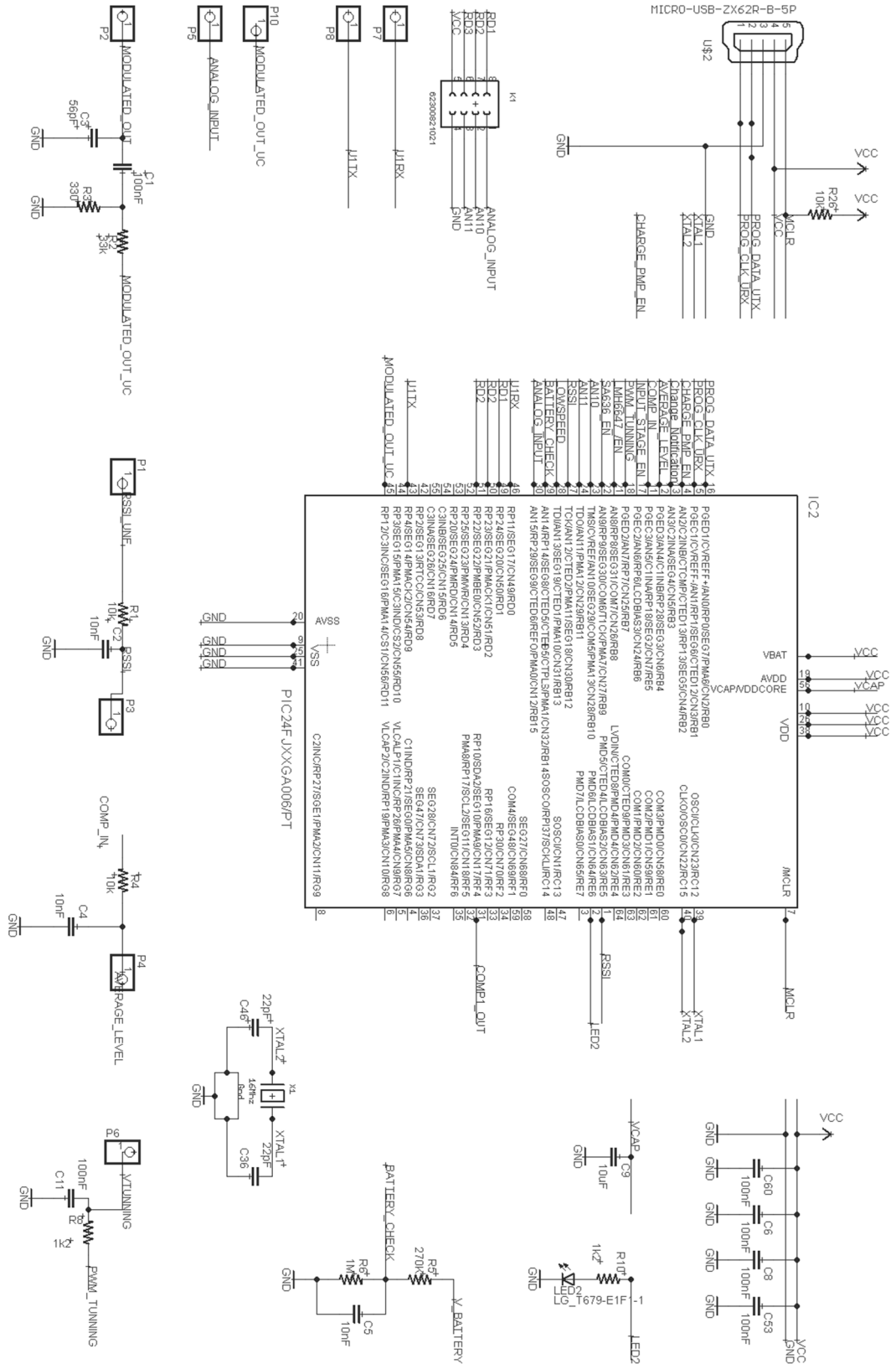


Figura A.1. Microcontrolador.



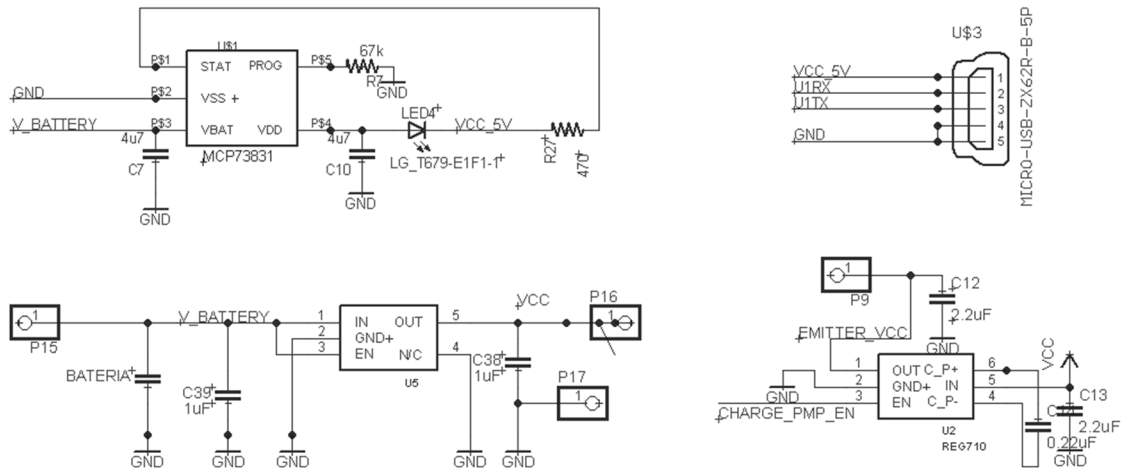


Figura A.2. Controlador de carga, regulador de tensão e multiplicador de tensão.

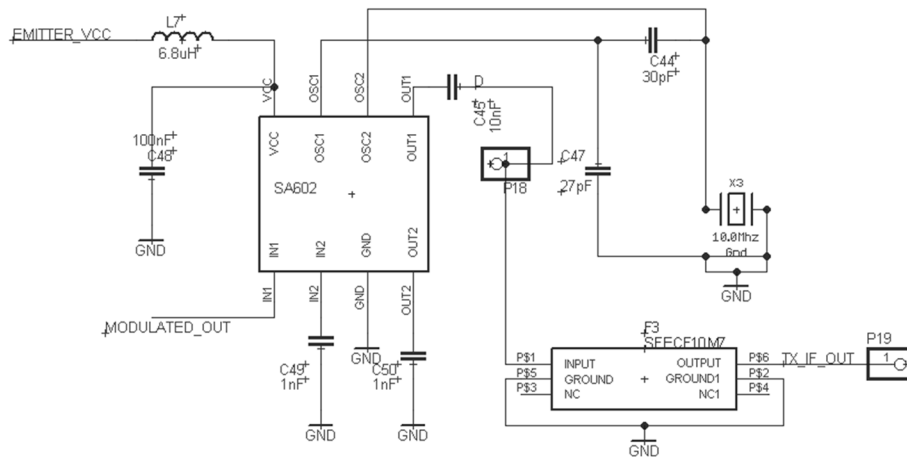


Figura A.3. IF *up-conversion mixer* e filtro passa-banda.

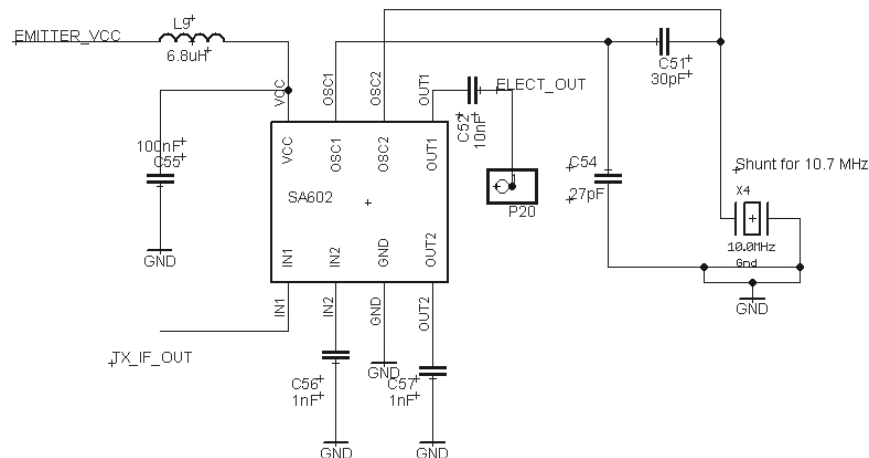


Figura A.4. RF *up-conversion mixer*.

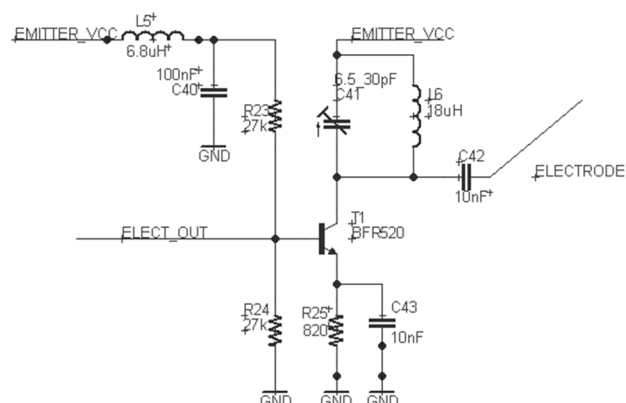


Figura A.5. Amplificador sintonizado do emissor.

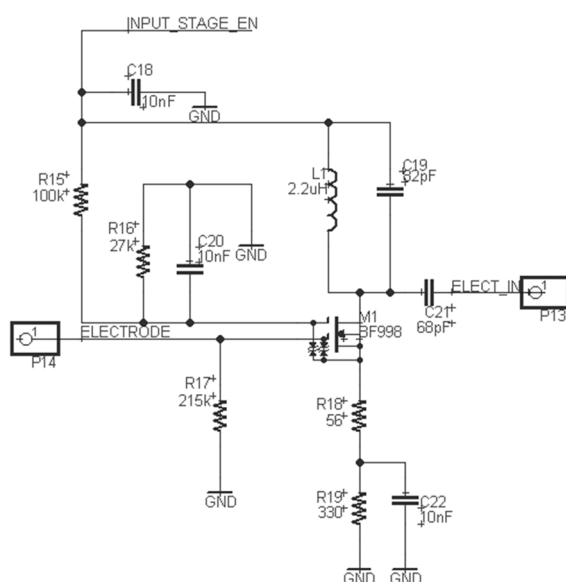


Figura A.6. Amplificador sintonizado do receptor.

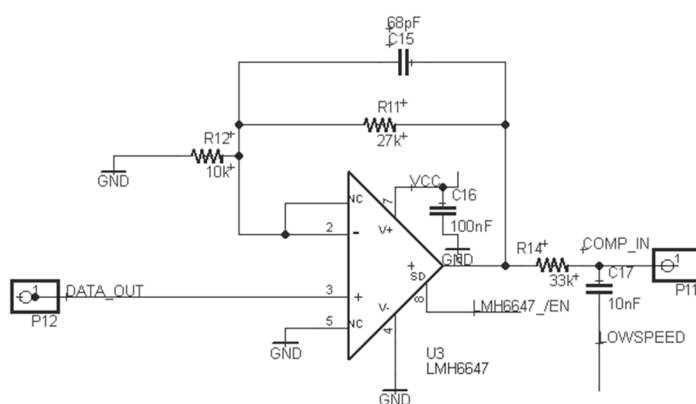


Figura A.7. Amplificador de banda-base do receptor.

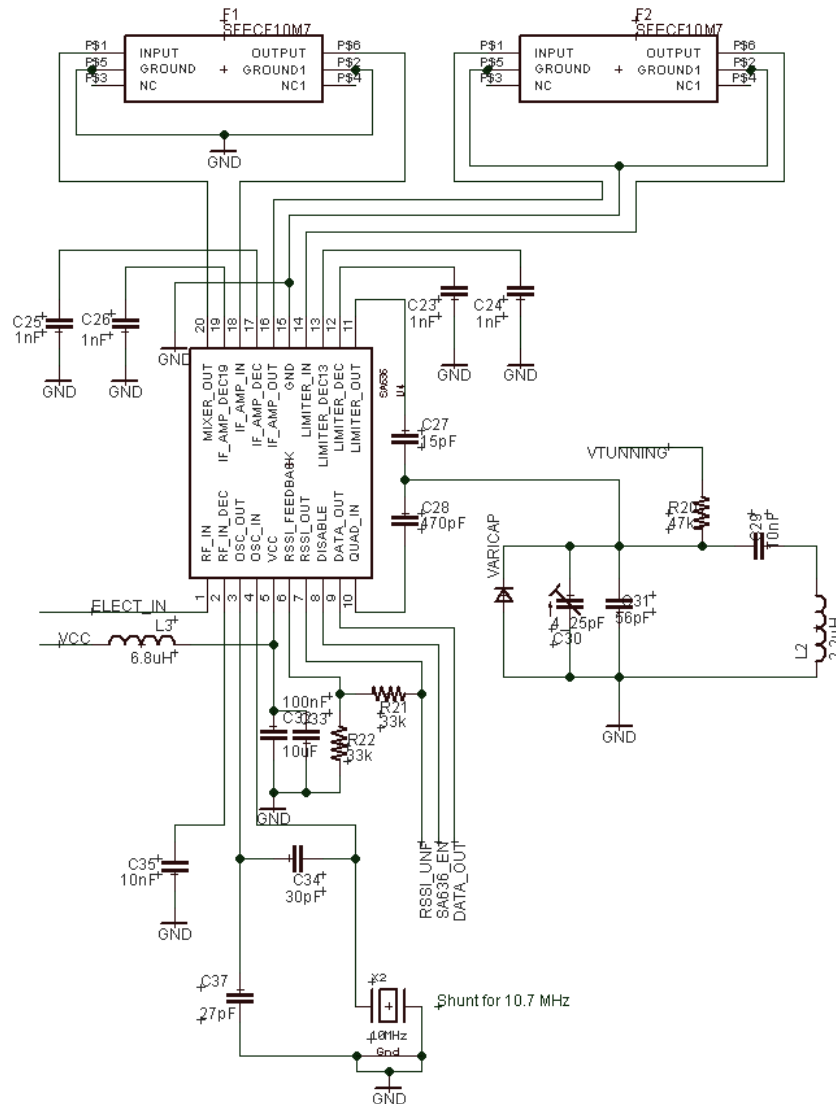


Figura A.8. Andar de FI do receptor.



## Apêndice B

# Esquemáticos da placa IOIO-OTG

